

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

REPRESENTING AND MANAGING CHAINS OF CUSTODY IN CYBER
FORENSICS USING LINKED DATA PRINCIPLES

THESIS

SUBMITTED

AS A PARTIAL REQUIREMENT

OF A DOCTORATE IN COGNITIVE INFORMATICS

BY

TAMER GAYED

OCTOBER 2016

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

REPRÉSENTATION ET GESTION DE CHAINES DE TRAÇABILITÉ EN
CYBERCRIMINALITÉ EN UTILISANT LES PRINCIPES DES DONNÉES LIÉES

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR

TAMER GAYED

OCTOBRE 2016

ACKNOWLEDGMENTS

I would like to thank my wife Dalia Habashy and my kids Anthony and Carol for all of their moral supports throughout this program. Family is so important, and you have been there every step of the way.

The first person I would like to thank at the university is my supervisor Hakim Lounis, who dedicated himself wholehearted to my research. He tried to understand my ideas and helped me to improve them by painstakingly reading this dissertation line by line and correcting my mistakes. I admire his patience, his comprehension, and his great support and devotion to me and my research. I also wish to express to him my most sincere gratitude for his emotional support.

Also, I want to thank my co-supervisor Moncef Bari, who accepted to work with me and supported my ideas. I wish to thank him for his great support, care, and patience toward my questions about cognitive field. Thank you very much.

Lastly, I would like also to thank all people who participated directly or indirectly in the realization of this work.

DEDICATION

*I dedicate this research to the late and beloved Mother Beatrice Sawares.
She devoted her whole life to me and my brother.
Words are not enough to express my feelings of gratitude to you.
You stood behind me through every success in my life.
Thank you, Mum. We will always remember you.*

FOREWORD

With the advent of digital technologies, tangible Chains of Custody (CoCs) that refer to the chronological documentation of physical or electronic evidence now need to undergo a radical transformation from documents to electronic representation. This is especially true in cyber forensics, where all evidence is of a digital nature. That fact requires judges to understand the field of Information and Communication Technologies (ICT), in addition to their legal knowledge.

This research proposes a novel framework to record, electronically, all information related to a cyber forensics investigation through the technology used by semantic web to create linked data. This technology is known as the web aspects.

The novel framework can help the technicians to record and represent the tangible chains of custody related to their investigation process in order to be provided to the judge in a court of law. It is assumed that the forensic information is prior improved, and the technicians who collected this information will use this framework to represent and transform the tangible documents into electronic data. Another level of improvements will be provided by this framework to annotate the forensic information using provenance vocabularies imported from the semantic web, and secure this represented information using Public-key Infrastructure (PKI).

On the other hand, this framework can also be applied to other domains, not only in the cyber forensics field. The cyber forensics domain is considered, in this dissertation, as a case study to explain how these aspects can be used and highlight various advantages of using such aspects to represent forensic information. In the

selected case study, such concepts are applied to represent the Chain of Custody documents generated from the forensic process.

The proposed framework can be applied in various domains, where information needs to be represented and need full traceability combined with a totally secure remote access in order to facilitate its consumption and ensure its confidentiality (i.e., understandable, descriptive, interlinked, and discoverable). For example, it can be applied in the domain of medicine, where doctors can record and represent different information about their patients. It can also be used by governmental agencies that address citizenship and immigration to record and manage information about citizens and immigrants, etc.

The actual dissertation discusses this framework as a first step to implement a system for role players (technicians, prosecutors and defendants) and judges to facilitate their legal procedures and help them to understand the digital evidence of cyber forensics presented to them in form of tangible documents. The framework contains a set of modules. Each module can be extended to accommodate different technologies provided recently by the field of Information and Technology (IT).

The work presented in this research does not implement all technologies. Instead, it opens the door for researchers who want to extend and enhance each module to add more and better technologies.

Each module of this framework touches on a certain discipline. For example, in the current dissertation, discussed is a module related to semantic vocabularies. It is based on some vocabularies imported from the semantic web; this does not mean that this module is limited to such vocabularies. More vocabularies from the semantic web can be imported to foster and extend the said module. Another module in the framework concerns provenance metadata, where a lot of work is related to this

sector of research. These works can also be used to extend and ameliorate the objectives of this module within the framework.

In addition, the PKI module of the framework is a module responsible to authenticate and bend the publication of data from an open to a closed scale. Further research may propose other security options to be added to the digital certificates used in the PKI.

Thus, what will be presented in this dissertation is a first version of this proposed framework. It opens the door to more extensions in a future work.

Nevertheless, the current version of this framework can serve technicians to record and represent their forensic investigation, and can be used by prosecutors/defendants or their attorney, and judges in a court of law to consume and understand the forensic information related to all digital evidence provided to them.

TABLE OF CONTENTS

FOREWORD	vii
LIST OF FIGURES	xvii
LIST OF TABLES	xxiii
ABBREVIATIONS AND ACRONYMS	xxv
RÉSUMÉ	xxix
ABSTRACT	xxxi
INRODUCTION	1
CHAPTER I	
RESEARCH PROBLEM.....	7
1.1 Introduction	7
1.2 CoC challenges	14
1.2.1 Accommodation with digital technologies.....	15
1.2.2 Fostering trustworthiness among role players and judges	16
1.2.3 Judges' awareness of digital evidence	17
1.2.4 Security of tangible CoCs information.....	18
1.3 Research hypotheses.....	19
1.4 Thesis organization.....	20
CHAPTER II	
STATE OF THE ART	23
2.1 Introduction	23
2.2 Accommodation with digital technologies	26
2.2.1 Semantic web and web of data.....	27
2.2.2 Cyber forensics and digital evidence	59
2.3 Fostering trustworthiness among role players and judges.....	68
2.3.1 Provenance vocabularies	71

2.3.2	Open Provenance Model (OPM).....	72
2.3.3	Named Graph (NG).....	73
2.4	Judges awareness of the digital evidence	76
2.4.1	Browsing pattern	77
2.4.2	Crawling pattern.....	78
2.4.3	Querying pattern.....	78
2.4.4	Reasoning pattern.....	79
2.4.5	Linked Education (LE).....	79
2.5	Security of COC information	80
2.5.1	PKI and digital certificates.....	82
2.5.2	Purposes and advantages	83
2.5.3	Protocols.....	84
2.5.4	Types and exchanges.....	85
2.6	Conclusion.....	86
CHAPTER III		
RESEARCH METHODOLOGY		89
3.1	Introduction	89
3.2	Representing COC using LDP.....	90
3.2.1	Why LDP for representing forensic information?.....	92
3.2.2	Correspondence between forensic phase and ontology.....	100
3.2.3	Creating proprietary terms	102
3.3	Adding provenance metadata to the <i>e</i> -CoC.....	103
3.4	Consumption patterns	107
3.5	Adapting Public-Key Infrastructure to Linked Opened Data (LOD)	108
3.5.1	Why use the PKI approach?	113
3.6	Proposed framework.....	115
3.7	The framework environment	119
3.8	Conclusion	120
CHAPTER IV		
CREATING AND PUBLISHING PROPRIETARY TERMS USING		

LIGHTWEIGHT ONTOLOGY AND ANNOTATING THEM USING PROVENANCE METADATA	121
4.1 Introduction	121
4.2 Selection of terms	123
4.3 Defining proprietary terms	125
4.3.1 Creation of ontology object (vocabulary)	125
4.3.2 Creation of new terms	128
4.4 Publication of proprietary terms	139
4.5 Annotation using provenance metadata.....	144
4.6 Conclusion.....	146
CHAPTER V	
CONSUMPTION PATTERNS	149
5.1 Introduction	149
5.2 Browsing and serializing	151
5.3 Crawling	163
5.4 Reasoning	166
5.5 Querying.....	170
5.6 Conclusion.....	172
CHAPTER VI	
LINKED CLOSED DATA USING PUBLIC-KEY INFRASTRUCTURE.....	173
6.1 Introduction	173
6.2 Creation phases.....	175
6.2.1 Creation of self-signed certificate	176
6.2.2 Creation of server certificate	177
6.2.3 Creation of client certificate.....	179
6.3 Installation of the digital certificates	180
6.3.1 Installation of self-signed certificate:.....	181
6.3.2 Installation of server certificate	181
6.3.3 Installation of client certificate.....	183
6.4 Working scenarios of the digital certificates	184

6.5	Heartbleed: an error in the OpenSSL tool	187
6.6	Conclusion	188
CHAPTER VII		
APPLYING THE CF-COC SYSTEM ON A COMPLETE FORENSIC PROCESS		191
7.1	Introduction	191
7.2	Identification of role players and judge	195
7.3	Publishing CoCs using CF-CoC	197
7.3.1	The acquisition phase	203
7.3.2	The authentication phase	215
7.3.3	The analysis phase	230
7.4	Adding provenance information to <i>e</i> -CoCs	245
7.4.1	Adding provenance metadata to the acquisition phase	246
7.4.2	Adding provenance metadata to the authentication phase	246
7.4.3	Adding provenance metadata to the analysis phase	247
7.5	Applying the consumption patterns on the <i>e</i> -CoC of Kruse model case study	248
7.5.1	Browsing the <i>e</i> -CoC of Kruse model case study	249
7.5.2	Crawling the <i>e</i> -CoC of Kruse model case study	252
7.5.3	Reasoning the <i>e</i> -CoC of Kruse model case study	253
7.5.4	Querying the <i>e</i> -CoC of Kruse model case study	267
7.6	Conclusion	268
CHAPTER VIII		
CONCLUSION AND FUTURE WORK		271
8.1	Introduction	271
8.2	Organization and scopes	272
8.3	Contribution to the computer science dimension	274
8.4	Contribution to the cognitive dimension	275
8.5	Limitations and future work	276
8.5.1	Framework modules	276
8.5.2	Semantic vocabularies	277

8.5.3 Creating ontologies for cyber forensics	277
8.5.4 Machine consumption	278
8.5.5 Linked Education	278
BIBLIOGRAPHY	279

LIST OF FIGURES

Figure	Page
1.1	Conceptual diagram of CoC 9
1.2	Abstract scenario of cyber investigation..... 12
1.3	Expansion of a cyber forensics term..... 17
2.1	Disciplines hierarchy of the state of the art 25
2.2	Web of documents 27
2.3	Web of data 28
2.4	RDF Models..... 35
2.5	Part of the Linking Open (LOD) Data Project Cloud Diagram 2014 39
2.6	Activity diagram of Kruse model 60
2.7	Use case diagram of Kruse model 61
2.8	RDF triples..... 74
2.9	Named Graph for RDF model 74
2.10	Graph identifier with metadata 75
2.11	Digital certificate 82
2.12	Sharing SSL/TLS certificates 85
3.1	Correspondence of forensic phases and LD resources 93
3.2	Interrelation between two datasets..... 94
3.3	RDF model for AFF4 vocabulary 99
3.4	Correspondence between cyber forensic phase and Ontology.....101
3.5	Named Graphs for the Kruse Model..... 105
3.6	Client/Server certificate exchange between two datasets 111
3.7	Use cases diagram of CF-CoC..... 116
3.8	CF-CoC framework 117

3.9	User interface of CF-CoC system	119
4.1	Activity diagram of creating and annotating proprietary terms	121
4.2	Screen for creating an ontology	126
4.3	RDF screen model of the acquisition ontology	127
4.4	Screen for creating a proprietary term	128
4.5	Screen for creating the “ <i>RolePlayer</i> ” class	130
4.6	Screen for creating the “ <i>FirstResponder</i> ” class	131
4.7	Screen for creating the “ <i>DigitalMedia</i> ” class	132
4.8	Screen for creating the “ <i>preserve</i> ” property	134
4.9	Screen for creating the “ <i>preservedBy</i> ” property	135
4.10	Screen for creating the “ <i>SN</i> ” property	136
4.11	T-Box of “ <i>SN</i> ” property	137
4.12	The T-Box of the “ <i>preserve</i> ” property	138
4.13	Class diagram of <i>e-CoC</i>	139
4.14	Screen for publishing RDF triples	141
4.15	Screen showing domain and range of a proprietary term	142
4.16	A-Box ontology of the forensic preservation task (<i>e-CoC_{Acq}</i>)	144
4.17	Screen for adding provenance metadata to the NG	146
5.1	Use case diagram of consumption patterns	149
5.2	Screen for consumption patterns	150
5.3	Screen for browsing	152
5.4	Screen of all forensic phases	153
5.5	Screen for viewing provenance vocabulary	154
5.6	<i>e-CoCs</i> for two forensic phases	155
5.7	RDF/XML of <i>e-CoC</i> for preservation task in the acquisition phase	156
5.8	Screen showing resources of preservation task in the acquisition	157
5.9	Screen for “ <i>FirstResponder</i> ” resource expansion	158
5.10	Screen for “ <i>RolePlayer</i> ” resource expansion	158

5.11	Screen for “ <i>Person</i> ” definition	159
5.12	Screen for “ <i>SN</i> ” resource expansion.....	160
5.13	Screen for RDF instances of “ <i>SN</i> ” property	161
5.14	Screen for “ <i>preserve</i> ” resource expansion.....	162
5.15	Screen for RDF instances of “ <i>SN</i> ” property	162
5.16	Screen for crawling resources/literals.....	164
5.17	Screen for crawling the “ <i>preserve</i> ” term	164
5.18	Screen for crawling using the “ <i>PDA device</i> ”	165
5.19	Screen for reasoning on a forensic phase	166
5.20	Screen for reasoning on “ <i>preserve</i> ” and “ <i>SN</i> ”	169
5.21	Screen showing all RDF triples	171
5.22	Screen for querying upon subject slot	172
6.1	Procedures for creating a digital certificate using the OpenSSL tool...	175
6.2	The CA self-signed certificate	177
6.3	The server digital certificate	178
6.4	The client digital certificate	180
6.5	Access to CF-CoC host server using HTTP	182
6.6	Access to CF-CoC host server using HTTPS	183
6.7	Redirection to the restricted resources	186
7.1	The tangible CoCs of the Kruse model.....	193
7.2	Screen showing some client certificates	196
7.3	Screen for creating “ <i>SuspectedDevice</i> ” class.....	203
7.4	RDF Model for “ <i>SuspectedDevice</i> ” class	204
7.5	Screen for creating “ <i>recover</i> ” property.....	204
7.6	RDF Model for the “ <i>recover</i> ” property	205
7.7	Screen for creating the “ <i>DeletedFiles</i> ” class	206
7.8	RDF Model for the “ <i>DeletedFiles</i> ” class	206
7.9	Screen for creating “ <i>containsRecover</i> ” property	207

7.10	RDF Model for the “containsRecover” property.....	208
7.11	Screen for creating the “ <i>backup</i> ” property	209
7.12	RDF model for the “ <i>backup</i> ” property	210
7.13	Screen for creating the “ <i>backupTo</i> ” property	211
7.14	RDF model for the “ <i>backupTo</i> ” property	212
7.15	Screen of publishing a triple using the “ <i>containsRecover</i> ” predicate ...	214
7.16	<i>e-CoC</i> of the acquisition phase	214
7.17	RDF/XML of the <i>e-CoC</i> of the acquisition phase	215
7.18	Screen for creating the “ <i>Authenticator</i> ” class.....	216
7.19	RDF model for the “ <i>Authenticator</i> ” class.....	216
7.20	Screen for creating the “ <i>PrimaryDevice</i> ” class	217
7.21	RDF model for the “ <i>PrimaryDevice</i> ” class	217
7.22	Screen for creating the “ <i>authenticatePrimary</i> ” property.....	218
7.23	RDF model for the “ <i>authenticatePrimary</i> ” property.....	219
7.24	Screen for creating the “ <i>ImagefilePrimary</i> ” class	220
7.25	RDF Model for the “ <i>ImagefilePrimary</i> ” class.....	221
7.26	Screen for creating the “ <i>hashingPrimary</i> ” property.....	221
7.27	RDF model for “ <i>hashingPrimary</i> ” property	222
7.28	Screen for creating the “ <i>checksumPrimary</i> ” property	223
7.29	RDF model for the “ <i>checksumPrimary</i> ” property	224
7.30	Screen for creating the “ <i>chckalgorithmPrimary</i> ” property	225
7.31	RDF model for the “ <i>chckalgorithmPrimary</i> ” property	225
7.32	<i>e-CoC</i> of the authentication phase	227
7.33	Screen for mapping the source device	228
7.34	Screen for mapping the backup device	228
7.35	<i>e-CoC</i> of acquisition and authentication phases	229
7.36	Screen for creating the “ <i>Analyzer</i> ” class	230
7.37	RDF model of the “ <i>Analyzer</i> ” class.....	231

7.38	Screen for creating the “ <i>analyze</i> ” property	231
7.39	RDF model of the “ <i>analyze</i> ” property	232
7.40	Screen for creating the “ <i>dataSize</i> ” property.....	233
7.41	RDF model of the “ <i>dataSize</i> ” property	233
7.42	Screen for creating the “ <i>analyzedBy</i> ” property.....	234
7.43	RDF model of the “ <i>analyzedBy</i> ” property	234
7.44	RDF model of the “ <i>totalSize</i> ” property.....	235
7.45	Screen for creating the “ <i>HiddenPartition</i> ” class.....	236
7.46	RDF model of the “ <i>HiddenPartition</i> ” class	236
7.47	Screen for creating the “ <i>contains</i> ” property	237
7.48	RDF model of the “ <i>contains</i> ” property	237
7.49	Screen for creating the “ <i>hiddenContains</i> ” property.....	238
7.50	RDF model of the “ <i>hiddenContains</i> ” property	238
7.51	Screen for creating the “ <i>hiddenUsing</i> ” property.....	239
7.52	RDF model of the “ <i>hiddenUsing</i> ” property	240
7.53	RDF/XML Code for an AFF4 format	241
7.54	<i>e</i> -CoC of the analysis phase.....	242
7.55	RDF model for an AFF4 result	242
7.56	<i>e</i> -CoC of the complete case study.....	244
7.57	Provenance graph for the acquisition phase	246
7.58	Provenance graph for the authentication phase	247
7.59	Provenance graph for the analysis phase	248
7.60	Screen for displaying the acquisition phase resources.....	249
7.61	Screen of the backup task in the acquisition phase.....	250
7.62	Screen for displaying the resources of the authentication phase	251
7.63	Screen for displaying the resources of the analysis phase	251
7.64	Screen for crawling using the keyword “ <i>Robert</i> ”	252
7.65	Screen for crawling using the keyword “ <i>LaptopHD</i> ”	253

7.66	Screen for querying using “ <i>Hard_drive_laptop</i> ”	268
------	--	-----

LIST OF TABLES

Table	Page
1.1 Thesis organization.....	20
2.1 RDFS Constructors for Property and Class Terms.....	42
2.2 OWL Constructors for Property and Class Terms.....	46
2.3 Rules and entailments of RDFS and OWL.....	55
2.4 Digital Forensics Process Models.....	60
3.1 Forensic Named Graph	105
3.2 Research problems and corresponding solutions.....	118
4.1 Proprietary terms of preservation task.....	124
7.1 Forensic terms of Kruse model	199
7.2 Classes of the acquisition phase.....	254
7.3 Properties of the acquisition phase	255
7.4 Classes of the authentication phase	257
7.5 Properties of the authentication phase	259
7.6 Classes of the analysis phase	264
7.7 Properties of the analysis phase.....	265

ABBREVIATIONS AND ACRONYMS

A-Box	Assertion Box
AFF	Advanced Forensic Format
CA	Certificate Authority
CDESf	Common Digital Evidence Storage Format
CF	Cyber Forensics
CF-CoC	Cyber Forensic – Chain of Custody Framework
CoC	Chain of Custody (tangible)
CoCs	Chains of Custody
CRC	Cyclic Redundancy Check
CVE	Common Vulnerabilities and Exposures
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DEB	Digital Evidence Bag
DEMF	Digital Evidence Management Framework
DFRW	Digital Forensics Research Workshop

xxvi

DSL	Digital Subscriber Line
<i>e</i> -CoC	Electronic Chain of Custody
<i>e</i> -CoCs	Electronic Chains of Custody
EJBCA	Enterprise Java Bean Certificate Authority
<i>e</i> -Justice	Electronic Justice
EWf	Encase Expert Witness Format
FCC	Federal Communication Commission
FDE	Federal Rules of Evidence
FM	Flowthing Model
FOAF	Friend of a Friend Vocabulary
FTP	File Transfer Protocol
GPS	Global Positioning System
GUID	Global Unique Identifier
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
ICT	Information and Communication Technology
JSON	JavaScript Object Notation
IP	Internet Protocol

LCD	Linked Closed Data
LDP	Linked Data Principles
LE	Linked Education
LSC	Legal Services Cooperation
MDA	Message Digest Algorithm
MMC	Microsoft Management Console
NFO	Nepomuk File Ontology
NG	Named Graph
NIE	Nepomuk Information Element Ontology
NIST	National Institute of Standards and Technology
OWL	Ontology Web Language
PC	Personal Computer
PHP	Personal Home Page
PKI	Public-key Infrastructure
RDF	Resource Description Framework
RDFID	Radio Frequency Identification
RDFS	Resource Description Framework Schema
RFC	Request for Comments

xxviii

SHA	Secure Hash Algorithm
SIN	Social Insurance Number
SKOS	Simple Knowledge Organization System
SPARQL	SPARQL Protocol and RDF Query Language (recursive acronym)
SSL	Secure Socket Module
SSN	Social Security Number
SWP	Semantic Web Publishing
SWSE	Semantic Web Search Engine
T-Box	Terminology Box
TEL	Technology-Enhanced Learning
TLS	Transport Module Security
URI	Unified Resource Identifier
URL	Unified Resource Locator
USDOJ	United State Department of Justice (USDOJ)
USB	Universal Serial Bus
US DOJ	United States Department of Justice National Institute
W3C	World Wide Web Consortium
XSD	XML Schema Data types

RÉSUMÉ

Les acteurs d'un processus judiciaire accumulent et enregistrent les informations et preuves accumulées durant leurs enquêtes, afin de les présenter à un juge dans une cour de justice. Lorsque ces informations sont enregistrées et consignées, elles constituent des artéfacts tangibles appelés chaînes de traçabilité (CoC). Les données fournies dans ces documents jouent un rôle vital dans le processus d'enquête, parce qu'elles répondent à des questions sur la façon dont les preuves sont collectées, transportées, analysées et conservées, depuis leur saisie jusqu'à leur production devant un tribunal. Des métadonnées de provenance accompagnent aussi ces données contenues dans les CoCs, afin de répondre aux questions sur leur origine et d'instaurer une confiance entre les différents acteurs du processus judiciaire, avec comme objectif ultime, le fait de rendre ces CoCs recevables devant une cour de justice.

Aujourd'hui, avec l'avènement de l'ère numérique, les enquêtes sont non seulement appliquées aux crimes physiques, mais font aussi référence à des preuves qui sont de nature numérique et peuvent ne pas être compréhensibles par des juges. Il en découle la nécessité que ces CoCs, documents tangibles, subissent une transformation radicale, du format papier vers des données électroniques, afin de tenir compte de cette évolution et de produire donc, des CoCs électroniques (*e-CoCs*), lisibles, compréhensibles et exploitables aussi bien par les humains que par les machines.

Le Web sémantique offre un cadre pertinent pour représenter et manipuler les CoCs, car il utilise des principes de Web connu sous le nom de Web des données (Principes des données liées, LDP), qui fournissent des informations utiles en RDF (Resource Description Framework, un modèle de graphe destiné à décrire de façon formelle les ressources du Web et leurs métadonnées), à travers des identifiants uniformes de ressources (URI). En outre, il comprend différents vocabulaires de provenance qui peuvent être utiles pour exprimer et promouvoir les métadonnées judiciaires. Ces principes sont utilisés pour publier les données publiquement sur le Web et donc proposer des données liées ouvertes, connues sous l'appellation de Linked Open Data (LOD). Cependant, l'aspect public des données d'enquêtes et de leurs métadonnées ne serait pas souhaité. Elles doivent obéir à certaines restrictions d'accès pour être partagées uniquement entre acteurs autorisés. Ces LDP peuvent être configurés pour publier des données sur une petite échelle, en utilisant l'approche de l'infrastructure à clé publique (PKI). Ainsi, la CoC représentée sera publiée sur une échelle restreinte

xxx

et ne pourra être consommée que par les acteurs concernés, à travers différents patrons de consommation de données.

Cette thèse fournit un cadre complet expliquant comment les CoCs et les données de provenances sont représentées et publiées en utilisant LDP, et comment l'infrastructure PKI peut être utilisée pour restreindre l'accès à ces données/ressources, afin d'être partagé à une échelle restreinte. L'évaluation de ce cadre se fera à travers une expérimentation empirique appliquée sur un modèle judiciaire complet.

MOTS-CLÉS: chaînes de traçabilité, cybercriminalité, données liées ouvertes, principes des données liées, patrons de consommation de données, infrastructure à clé publique.

ABSTRACT

Role players of any forensic investigation process record chronologically all forensic data resulted from their investigation in order to be presented to the judge in a court of law. When these results are recorded and posted, they are called Chains of Custody (CoCs). The forensic data provided within these documents play a vital role in the process of forensic investigation, because they answer questions about how evidence is collected, transported, analyzed, and preserved since its seizure until its production in court. Provenance metadata also accompany these forensic data to answer questions about their origin and foster trustworthiness among role players and judges in order to make the tangible CoCs admissible in a court of law.

Nowadays, with the advent and evolution of the digital age, the forensic investigation is not only applied to physical crimes, but also to digital evidence and may not be understandable by judges in the courts of law. This fact increases the need that these tangible documents undergo a radical transformation from paper to electronic data in order to accommodate this evolution and provide electronic-CoC (e-CoC) readable, understandable, and consumable by humans and machines.

The semantic web is a fertile land to represent and manage tangible CoCs, because it uses web principles known as Linked Data Principles (LDP), which provide useful information in Resource Description Framework (RDF) format upon Unified Resource Identifiers (URI) resolution. In addition, it includes different provenance vocabularies that can be useful to express and foster the forensic metadata. Generally, the power of LDP resides in publishing data publicly without any access restriction on the web. However, the openness of cyber forensics data and their metadata would not be convenient. Cyber forensics data should obey some access restriction in order to be shared only among role players and judges. Public-key Infrastructure (PKI) can be applied to restrict the access to some or all resources of the represented data and bends the LDP from open to closed consumption, while maintaining the resolution of such restricted resources. The judge will in turn consume the restricted represented data using different LDP consumption patterns. A role player can also be the consumer of such represented resources published by other role players.

This thesis provides a complete framework explaining how forensic and provenance data are represented and published using LDP, and how PKI can be used to restrict the access to these data/resources in order to be shared on a closed scale. The

evaluation of the framework will be done through an empirical experimentation applied in a complete forensic model.

KEYWORDS: Chains of Custody, Cyber Forensics, Linked Open Data, Linked Data Principles, Consumption Patterns, Public-key Infrastructure.

INTRODUCTION

The history of forensic investigation tasks dates back thousands of years. These tasks deal with gathering and examining evidence about the past in order to prosecute criminals in the future. With the advent of Information and Communication Technology (ICT), forensic investigation is not only concentrated on physical crimes, but also on digital evidence. A new type of forensic investigation, known as computer/cyber digital forensics, has emerged.

One of the most essential parts of the digital forensic process is the Chain of Custody (CoC). CoC is a chronological document accompanying all digital evidence, in order to avoid later allegations of tampering with such evidence. CoC provides useful information about the digital evidence produced during a forensic process by answering the five “Ws” and one “H” questions. The five “Ws” ask “When,” “Who,” “Where,” “Why,” and, “What,” while the “H” asks “How.”

Today, cyber forensics is a daily growing field that requires accommodation for the continuous changes in digital technologies. The tangible CoC information also needs to undergo a radical transformation from paper to electronic data (*e-CoC*), which is readable and consumable by computers.

The semantic web is a fertile land to represent this information because it is rich with different vocabularies and provenance metadata that can be useful to represent and manage such forensic information.

Nowadays, the semantic web is the web of data. However, it is not just concentrated on the interrelation between web documents, but also between raw data within these

documents. This data interrelation is based on four aspects provided by Tim Berner-Lee in 2006 known as the Linked Data Principles (LDP). These aspects allow the data being represented to be published in a structured way that can facilitate their consumption.

This dissertation provides a novel framework that uses the LDP to represent the tangible CoC in order to be consumed in a court of law. The framework provided in this dissertation is elaborated through a set of modules. Generally, it presents how the semantic web and its technologies presented in vocabularies and metadata are a fertile land to represent the tangible CoCs from their publication by the role players throughout the cyber forensics investigation, until their consumption by judges in a court of law.

In addition, the framework provided in this dissertation uses a Public-key Infrastructure (PKI) to ensure the identity and the authentication of each technician participating in the investigation process, and to protect and foster the published information related to the case in question from unauthorized access. This idea argues that not all information published on the web of data should be on an open scale. However, LDP need to be bent and adapted for publishing data with access restrictions in order to be shared on a closed scale. This is known as the Linked Closed Data (LCD).

This thesis is organized as follows: Chapter 1 discusses the research problems. Chapter 2 presents the state of the art. Chapter 3 concerns the research methodology. Chapters 4, 5, and 6 discuss the proposed framework. Chapter 7 applies the system to a complete forensic process. Finally, Chapter 8 summarizes the thesis and presents future prospects for this work.

The main benefits of this thesis comprise of transforming the tangible CoC into electronic CoC to:

- Accommodate with the new technology of knowledge representation;
- Foster trustworthiness among role players and judges by adding provenance metadata imported from the semantic web;
- Help judges understand and consume the electronic CoC using different consumption patterns; and
- Secure the represented resources in order to be used on a closed scale using the PKI.

All ideas behind this thesis are published and discussed in different international conferences. Thus, the following paper introduced the idea of using the semantic web to produce an electronic chain of custody:

Gayed, T. F., Lounis, H. et Bari, M. (2012a). Computer forensics: toward the construction of electronic chain of custody on the semantic web. International conference on software engineering and knowledge engineering, 406-411.

In the same year, 2012, we published the complementary work of the above paper. In this work we introduced the benefit of using the semantic web is to improve the CoC:

Gayed, T. F., Lounis, H. et Bari, M. (2012b). Cyber forensics: representing and (im) proving the chain of custody using the semantic web. International conference on advanced cognitive technologies and applications, 19-23.

In the following 2013 paper, we mention explicitly all advantages and rewards of using the LDP and the semantic web vocabularies to accommodate the tangible documents within the current era of technology:

Gayed, T. F., Lounis, H. et Bari, M. (2013b). Cyber forensics: representing and managing tangible chain of custody using the linked data principles. International conference on advanced cognitive technologies and application, 87-96.

On the other hand, in another paper, we discuss the steps to represent forensic resources applied to the preservation task of the Kruse model:

Gayed, T. F., Lounis, H. et Bari, M. (2013a). Representing chains of custody along a forensic process: a case study on Kruse model. International conference on software engineering and knowledge engineering, 674-680.

The year after, in 2014, two papers were published. The first one explains how to create custom terms using the Resource Description Framework Schema (RDFS). This is illustrated by using the acquisition phase imported from the Kruse model as a case study:

Gayed, T. F., Lounis, H. et Bari, M. (2014b). Creating proprietary terms using lightweight ontology: a case study on acquisition phase in a cyber forensics process. International conference on software engineering and knowledge engineering, 76-81.

The second paper is about how the PKI is exploited to publish data using LDP on a closed scale:

Gayed, T. F., Lounis, H. et Bari, M. (2014a). Linked closed data using PKI: a case study on publishing and consuming data in a forensic process. International conference on advanced cognitive technologies and applications, 77-86.

In 2015, a journal publication combined all ideas together has been released. It illustrates a complete scenario of using the LDP to publish and consume forensic resources, on a closed scale, using the PKI approach:

Gayed, T. F., Lounis, H. et Bari, M. (2015). Representing and Publishing Cyber Forensic Data and its Provenance Metadata: From Open to Closed Consumption. International Journal on Advances in Intelligent Systems, 7(3&4), 662-688.

Finally, while recent, our work is already referenced by some authors in the field namely, in journals such as “*Elsevier Digital Investigation*” and “*International Journal of Computer Applications*.”

CHAPTER I

RESEARCH PROBLEMS

1.1 Introduction

The field of Computer/Digital forensics is growing on a daily basis. It combines computer science concepts with evidentiary rules and legal standards to prosecute criminals of digital evidence in a court of law (Casey, 2014). At the most basic level, the digital forensic process has three major phases: acquisition, authentication, and analysis (Kruse II et Heiser, 2001; Köhn et al., 2008). Simply said, the acquisition is a phase where evidence is collected and extracted from the suspected digital devices (e.g., laptop, mobile phones, etc.). Authentication is the phase that ensures that the collected evidence is not altered and keeps its integrity. The analysis phase takes the acquired images to analyze and identify them into pieces of evidence in order to draw conclusions.

In an adversarial system there usually exist two advocates representing their parties' positions before a jury or judge. The parties are the state and the accused. The evidence related to a crime is usually collected by the police or by a party whose services were retained by the police and handed over to the prosecution. The prosecutor will decide which pieces of evidence to present at trial while the accused will try to contest the validity/eligibility of a given piece of evidence.

The information collected by an authorized party is recorded chronologically in tangible documents. These documents will be called tangible Chains of Custody (CoCs), because they keep track and provide useful information related to the collected evidences by answering the five “Ws” and one “H” questions.

A CoC is one of the most essential parts of any forensic investigation process (Ballou, 2010). It accompanies all digital evidences in order to avoid later allegations of tampering. Thus, the Chain of Custody is essential in this context.

The following points summarize the process since the collection of evidence until the judge’ engagement:

1. Technicians whose services were retained by the authorities of the police investigation (e.g., first responders, expert witnesses, police officers, bailiffs and investigators) will proceed with the collection of evidence;
2. The collected evidence will be forwarded to the prosecutor who will choose which elements to present at trial;
3. The evidence will be then disclosed to the accused or his attorney;
4. If the accused disputes the addressed evidence, a session will held to verify the admissibility of the collected evidence;
5. In this session, the judge will hear the testimony of technicians to determine if there were flaws in the chain of custody imposing rejection of evidence.

From the above process, we have four actors: technician, prosecutor, defender/accused and judge. In this dissertation, the first three actors will be called role players, and the fourth actor will be the judge.

Figure 1.1 depicts the conceptual diagram of a tangible CoC. The forensic process defines the role of technicians. The forensic process presented in this figure contains three phases exist in any forensic investigation: acquisition, authentication, and

analysis. Three types of technicians are needed to work on these forensic phases: first responder, authenticator and analyzer.

The main role of technicians is to collect evidence. Each technician may participate to one or more forensic phase(s). The simplest case, each technician is assigned to one forensic phase. In each phase, a technician can accomplish one or more forensic task(s) and each task in a forensic phase can be accomplished by one or more technicians. Each technician is responsible for creating his own chain of custody for each task assigned to him. Creating a chain of custody means that a technician records chronologically all collected evidence in tangible documents.

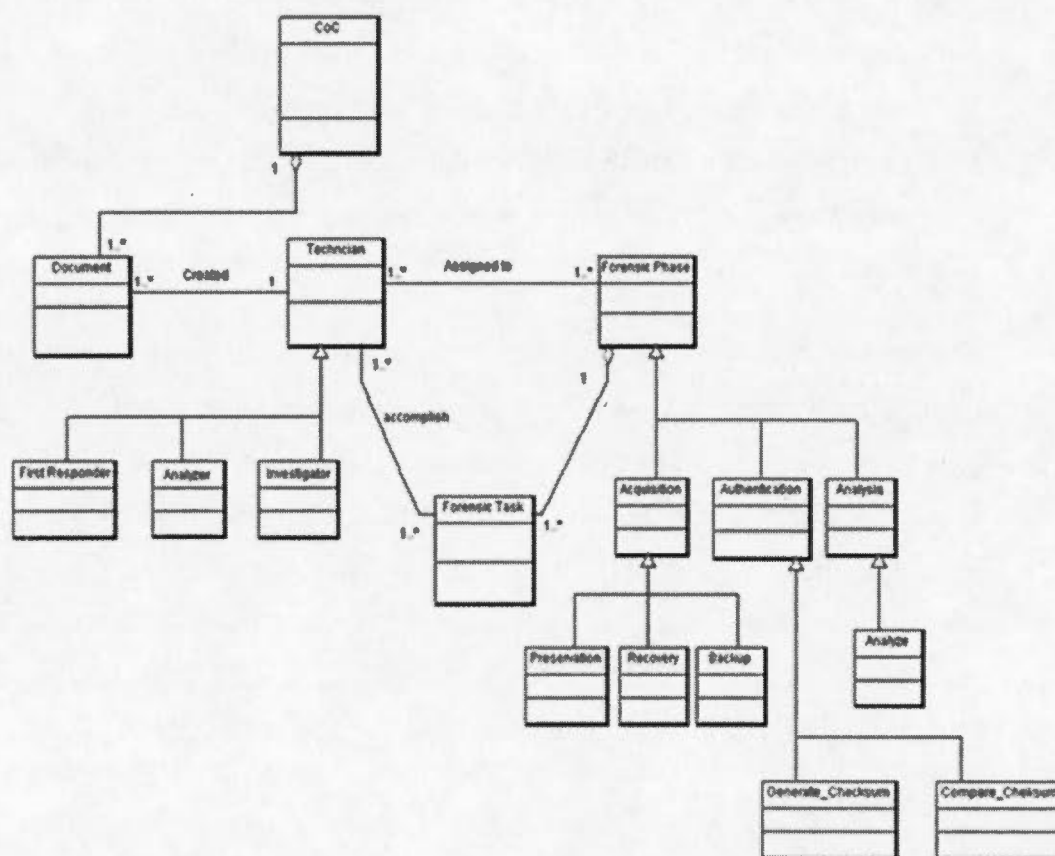


Figure 1.1 Conceptual diagram of CoC

This dissertation depicts how the tangible documents of the chain of custody will be transformed into electronic information consumable by human and machines. Chapter 7 presents a complete case using the Kruse model, which contains three forensic phases. Each phase is assigned to a technician. Technicians will use a system called CF-CoC, as proposed in this dissertation, to publish such information and annotate them using different provenance metadata. Role players (technicians, prosecutor, and defendants or their attorney) and judge will be able to consume the published/electronic information. In addition; the system applies a secure approach to restrict the access and consumption to the published resources using digital certificates.

Because the field of cyber forensics is growing on a daily basis and requires the accommodation for the continuous changes in digital technologies, the tangible CoC information also needs to undergo a radical transformation from paper to electronic data (*e*-CoC), which is readable and consumable by computers. This transformation is achieved by representing the information in a form that is understandable and that can be processed by computers.

The semantic web will be a flexible solution to achieve this goal, because it is based on an infrastructure that provides a means for publishing vocabularies that can be read by humans and processed by machines. This infrastructure is called “Resource Description Framework” (RDF) (Beckett et McBride, 2004). It allows the encoding, exchange, and reuse of structured metadata. It also contains several semantic markup languages developed under the auspices of the World Wide Web Consortium (W3C), such as RDF Scheme (RDFS) (World Wide Web Consortium (W3C), 2014) and Web Ontology Language (OWL) (McGuinness et Van Harmelen, 2004), which are based on XML (Bray et al., 2008). RDF data models are semantically encoded using RDFS and OWL (Berry et al., 2003).

Today, the semantic web is the web of data, which is not just concentrated on the interrelation between web documents, but also between the raw data within these documents. This data interrelation is based on standardized web technologies: (Berners-Lee et al., 2001; Campbell et MacNeill, 2010) the HTTP (Fielding, 2014), URI (Berners-Lee et al., 2014), and RDF (Beckett et McBride, 2004). It is designed to represent information in a machine-readable format by introducing different representation languages based on XML (Bray et al., 2008) or JSON¹ (JavaScript Object Notation), more recently. The latter is a lightweight data-interchange formats easily to understand by humans and machines. These technologies allow publishing structured data, so that it can be interlinked and navigable between each other.

In addition, the author will consider the semantic web as a fertile land for representing and describing the tangible CoCs since it is rich with different provenance vocabularies that are useful to describe the forensic information. These vocabularies can provide answers to the five Ws and one H questions related to the origin of this data. Some examples of widely deployed provenance vocabularies are the Dublin Core (DC) (Dublin Core Metadata Initiative, 2015), Friend of a Friend (FOAF) (Brickley et Miller, 2014), and Ontology Metadata Vocabulary (OMV) (Hartmann et al., 2005), which contain different predicates that can elaborate the published data with extra information and metadata.

Thus, representing forensic information using the technologies of the semantic web will be useful for each technician in the forensic process. It will allow technicians to record and publish their forensic investigation results in a structured and unified format. Furthermore, publishing data in such a format can facilitate the consumption of these data by the prosecution, defense and the judge in a court of law. This point will be discussed in detail in Chapters 2 and 3.

¹ <http://www.json.org/>

When a user performs an action using any digital device, such as computer, laptop, digital camera, smart phone, etc., if this action is committed in violation of a law, it is an illegal act and can be considered an infraction or even a crime, and its evidence is of digital nature. The performer of this act is then called the perpetrator of the crime. Some acts are country dependent. This means that the committed acts can be considered crimes in some countries and not in others.

Any discovered crime should obey a scientific investigation using certain forensic processes (Köhn et al., 2008). This investigation aims at gathering and examining information about the past in order to be used in the future in the court of law. Each phase in a forensic process is assigned to a person that is qualified to play a certain role in the forensic investigation, such as first responders, expert witnesses, police officers, bailiffs and investigators. (e.g., the first responder is a person who is qualified first-hand to seize, preserve and collect all the necessary information on the crime scene). The role players may be technician, the prosecution or the defense. One or more technicians may be assigned to a forensic phase. Technicians are the ones who are responsible to record and save all investigation results that are achieved during their forensic phase. Once they finish their tasks, they provide the collected information to the prosecutor to choose which elements to put into evidence as part of his prosecution.

For simplicity, Figure 1.2 considers that each forensic phase is assigned to a technician that creates his own CoC, describing all forensic information he collected and all results he deduced using different forensic tools.

The classical way used by technicians is to record their information in tangible documents, seal them in an envelope, and provide them to the prosecutor. The latter selects and prepares the evidence that he wants to use to prosecute the accused. The selected evidence will then be disclosed to the accused or his attorney. The recording task is usually performed manually and does not include any computers, except for

some classic tasks (e.g., word processing, printing and filling in documents, using emails, etc.).

The novel proposed system will reside somewhere on a web cloud and will be owned for instance, by a neutral side, which is not on conflict of interest with the prosecutor and defender, to facilitate the cyber investigation and justice process. The proposed framework will use web technologies to aid (i) the technicians to securely record (i.e., publish/represent) electronically the information related to their CoCs, (ii) prosecutor/defender to consume this information and append his prosecution elements and communicate with the defender/accused and finally, (iii) in case of dispute the judge to consume the represented information.

Once the technicians transform the tangible CoCs from paper to electronic form, the transformed (i.e., represented) information should obey some access restrictions, especially since the information being published will be shared in a public manner, which is due to the nature of the used aspects (i.e., LDP) of the web. Thus, the framework should provide a secure way to let the technicians publish their information, and after they finish their task, they should be able to share the published information with the prosecutors.

1.2 CoC challenges

As mentioned, CoCs documents record all information related to digital/physical evidence. They are also known as testimony documents, since they ensure and guarantee that all evidence related to the crime case in hand are not altered throughout the forensic investigation. Failure to record enough information related to the evidence may lead to its exclusion from legal proceedings. Furthermore, if the CoCs are not well-maintained and the suspect is guilty, the defense can argue that the

CoCs were not properly established and cast doubt on the acquired evidence (Casey, 2014). In such a case the judge will need to hear the testimony of the technicians to determine if there were flaws in the chains of custody imposing their rejection.

1.2.1 Accommodation with digital technologies

Today, technicians are still providing the forensic information describing their investigation process with the information that resulted from various forensic tools in the form of tangible documents. Most of the evidence manipulated in the digital forensics field is of digital nature. Even if the digital evidence and their investigation results are provided on digital devices and have a digital format, their testimony descriptions are provided within tangible documents. This is due to the fact that the audiences of these documents, in all countries, are prosecutors, defenders and judges who are mainly competent in the legal field. Thus, these documents should take an appropriate form that accommodates the audience receiving them.

Since the 1990s, the US DOJ (Department of Justice) National Institute has been trying to encourage and support all research that can prevent crimes and can improve criminal policy justice and practice (Sherman et al., 1997; Losavio et al., 2006).

All information describing the investigation process such as those that are recorded by technicians and those that resulted from forensic tools need to be unified and stored together to facilitate their interoperability and consumption.

In addition, all forensic information resulting from a forensic phase and published by a technician should be interlinked with other forensic information resulted from another forensic phase and published by another technician. This is because the forensic information is a co-operative task, where all technicians should participate

together in order to draw dependent conclusions. The investigation of a technician player may depend on the results and conclusion drawn by another technician.

1.2.2 Fostering trustworthiness among role players and judges

The problem is not only to represent the information included in tangible CoCs to solve the issues mentioned above. It is also to express information about where the CoC information came from. Judges can find the answers to their questions in the CoC, but they need to also know the provenance and origins of those answers. Provenance of information is crucial to guarantee and ensure the trustworthiness and confidence of the shared information among role players (technicians, prosecutors and defenders) and judges. Trustworthiness starts by identifying actors in order to build a secure channel to share information. However, when the objective is mainly focused on the information itself, the latter should be illustrated through various provenance information.

Hence, this dissertation will distinguish between forensic information and provenance information (Gayed et al., 2012a). Forensic information should be responsible to answer the five Ws and one H questions related to the case in hand, while provenance information should be responsible to answer questions about the origin of these answers (i.e., what information sources were used, when were they updated, how reliable the source was). This will be accomplished through the use of different provenance metadata of the semantic web. Providing answers to such questions fosters the trustworthiness among publishers (i.e., role players) and consumers (i.e., judges and role players) and makes the *e*-CoC admissible in a court of law.

1.2.3 Judges' awareness of digital evidence

In the literature, several works have been provided to identify the judges' understanding of ICTs underlying digital evidence (Losavio et al., 2006; Insa, 2007; Rogers et al., 2007). These works aim to design specialized training and education programs to ameliorate their ICT knowledge and make them better able to evaluate scientific and technical evidence presented in a court of law.

There exists different terms related to physical evidence, which are well-known by anyone. For example, the word "gun," is a common word that does not need to be interpreted and explained to understand. However, this is not the case in cyber forensics. In cyber forensics, terms are mostly related to information technology and such terms need to be interpreted and expanded to get and understand their meaning (see Figure 1.3).

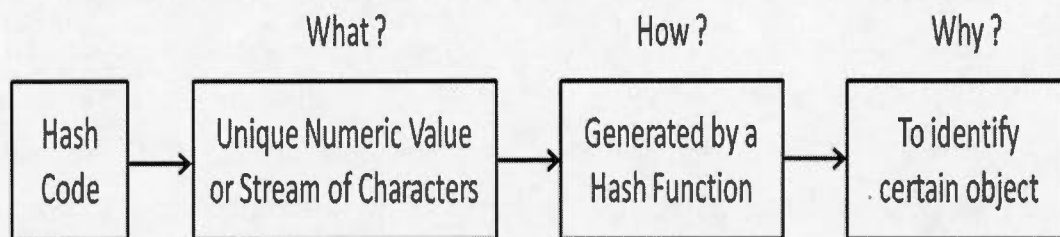


Figure 1.3 Expansion of a cyber forensics term

For example, as shown in Figure 1.3, a term called "*Hash Code*" is expanded to provide what is meant by a hash code, how it is generated, and why it is used. Along this path we may have another expansion path. For example, the How rectangle can be nested with more information to explain what a hash function is, and so on. This is called "linked information" or "descriptive information", where each piece of information can be dereferenceable to provide more information.

Judges are more specialized in, as well as better understand the legal domain and procedures than the field of ICT. Most judges do not have the required knowledge concerning information technologies, or the knowledge that they currently have is not enough to evaluate and take the proper decision for the case in question. One solution that has been proposed in the literature is to organize a training program to teach judges about the field of ICT (Kessler, 2010).

The dissertation argues against this solution's direction, because it will not be an easy task to teach judges the different concepts of ICT. Instead, it discusses the solution of providing a descriptive *e*-CoC, where technicians can publish the results of their investigation. The prosecutor, defender and judge (in case of dispute or contest from the defender) can consume such results in a descriptive and understandable way and thus take the proper decision regarding the case in question.

While consuming the represented information, a judge should have the ability to consume forensic information in a descriptive way. It is important that judges can find and discover more information about different resources, especially those that are unknown to them. LDP use the URI to identify different resources (i.e., subjects or things). These resources should have dereferenceable nature. This means that each resource can be expanded to other resources in order to get more information and navigate among other related resources. Representing the data using such a structure is an example of a new area of research called Linked Education (LE).

1.2.4 Security of tangible CoCs information

Usually, the CoC documents must be affixed securely when they are transported from one place to another. This is achieved in a very classical way: seal them in plastic bags together with physical evidence if there is any, such as a hard disk, USB, cables,

etc.), label them, and sign them into a locked evidence room with the digital evidence and devices themselves to ensure their integrity. The e-CoCs also need to be secured, from their publication by the technicians until their consumption by judges. LDP are used to publicly publish the data on the web and need to be adapted and bended with access restrictions.

1.3 Research hypotheses

This dissertation will verify a set of hypotheses. Each hypothesis is related to a research problem. The following hypotheses are mentioned in this section in the same order as research problems.

Hypothesis 1:

The semantic web can be a fertile land to create interlinked e-CoCs, which are readable and consumable by people and machines, and the forensic information resulting from a forensic tool can be interoperable with these interlinked CoCs.

Hypothesis 2:

Provenance metadata expressed in the formats used by the semantic web can be useful to answer the questions about the origin of the CoC data, and then foster trustworthiness among role players and judges.

Hypothesis 3:

Representing the CoC resources using the linked data principles can provide a descriptive e-CoC and then improve the subject matter and the understanding of the digital evidence.

Hypothesis 4:

PKI can be applied to the Linked Data (LD) to securely publish and consume the data among role players and judges, as well as transform the open data to closed data.

This dissertation will discuss a novel framework that solves the research problems mentioned in Section 1.2 and verifies the correctness of the proposed hypotheses by applying this proposed framework to a complete forensic process. The framework is named Cyber Forensics-Chain of Custody (CF-CoC). Before moving on to Chapter 2, the next section depicts how this thesis is organized with respect to the research problems and hypotheses.

1.4 Thesis organization

This section summarizes the organization of this thesis. Most of this organization depends on research problems and hypotheses order.

Table 1.1 Thesis organization

Chapters	CoC Challenges (Research problems)	Hypotheses	Proposed Solutions
Chapter 2	State of the art corresponding to challenges, hypotheses, and solutions.		
Chapter 3	Research Methodology		

Chapter 4	Accommodation with digital technologies	Hypothesis #1	Representing CoC using LDP
	Fostering the trustworthiness among role players and judge	Hypothesis #2	Adding provenance metadata to the <i>e</i> -CoC
Chapter 5	Judges awareness about the digital evidence	Hypothesis #3	Consumption patterns
Chapter 6	Security of CoCs information	Hypothesis #4	Adapting PKI to LOD
Chapter 7	Applying the CF-CoC system to a complete forensic model		
Chapter 8	Conclusions and future work		

The above table contains four columns: the first is the chapter number (i.e., excluding the current chapter), the second column indicates the challenges that encounter the tangible CoCs (i.e., excluding Chapter 2, which explains the state of the art of this research, and Chapter 3, which illustrates the research methodology). The third column is about the hypotheses of this research, of which there are four. Chapter 4 provides the solution related to the first two hypotheses. Chapter 5 is dedicated for the third hypothesis, and finally, Chapter 6 depicts the PKI to answer the last hypothesis.

CHAPTER II

STATE OF THE ART

2.1 Introduction

Technology and law are two different fields, but with the rapid evolution of technology occurring today, both fields are increasingly becoming related to one another and intersecting (i.e., marriage of law and technology). The convergence may occur from two scopes, whether the law converges with technology or technology converges with law.

First scope, if law converges toward technology, then we are going to exploit the law and its power to provide legal and secure electronic services for people who use the technology in their daily lives. This is known as “IT Law”, and consists of law and legal aspects to govern the information technology. The “IT law” is not the same of “IT aspects of law itself”. The latter are used to deliver legal services to people through the IT field.

Second scope, if technology converges to the law, then we are going to discuss what the technology can do to facilitate and enhance the legal procedures over all levels of a judicial system. This is known today as the field of electronic justice (*e-justice*).

Many works have been provided in literature on both scopes. For example, in the first scope, a work presented in (Yoo, 2005) discussed how the US Supreme Court cleared the way to the Federal Communications Commission (FCC) to resolve how to fit the

leading broadband technologies, such as Digital Subscriber Line (DSL) services and cable modems. Another example is provided in the field of *e-commerce*, where all *e-transactions* should be protected and controlled by legal procedures and law².

On the other hand, in the second scope, the work presented in (Cabral et al., 2012), discussed the use of technology to enhance access to justice. In this work, the authors discussed the enhancement of delivering of legal services to all private and public sectors of the United States. The Legal Services Corporation (LSC), which is responsible for these services, intended to increase the quantity and quality of services through technology, by developing web-based business processes using smart phones.

Another example of using technology to serve the law is the use of statistical data-mining techniques to detect credit card fraud (Chan et al., 1999) and using anomaly detection methods to identify potential terrorist activities (Seifert, 2004).

Researches related to both scopes are numerous and are not limited to those examples. Other examples can be found in the journal of technology and law³ to name but one source.

This thesis lies under the second scope, where the technology of the semantic web will be exploited to, (i) aid role players to maintain and represent the CoC, and, (ii) help judges understand digital evidence.

The state of the art related to the proposed framework goes over different disciplines, such as semantic web, Cyber Forensics (CF), provenance of information, and security. Therefore, the state of the art in this chapter will have different facets. Each facet discusses the related works of each discipline apart.

² <http://www.hg.org/ecommerce-law.html>

³ <http://jolt.law.harvard.edu>

The main classification of the state of the art, as shown in Figure 2.1, is organized according to the research problems that were proposed in the first chapter.

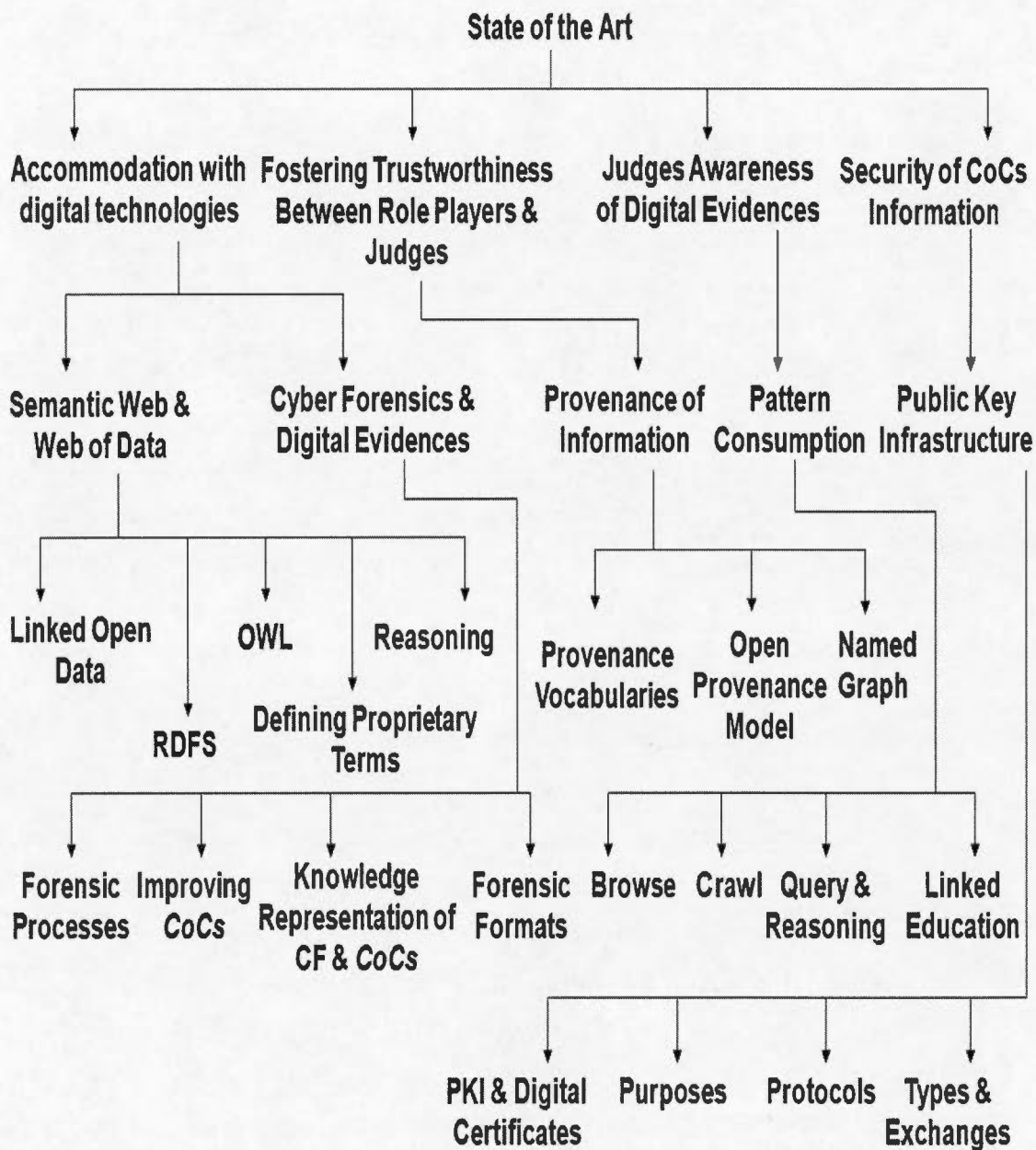


Figure 2.1

Disciplines hierarchy of the state of the art

This dissertation discusses four problems. Each problem is related to one or more discipline(s). For example the “Accommodation with digital technologies” is a problem related to two different disciplines: “The semantic web and web of data” and “The CF and digital evidence”. Each discipline contains a set of related works. Another example is the problem of “Fostering the Trustworthiness among Role players and Judges”. It is related to the discipline of “Provenance of information”, which can be used to foster the admissibility of chains of custody in a court of law. Under this latter discipline, different models are expanded to depict the recent published works related to such discipline.

The discipline(s) related to each problem is/are mentioned according to their utility in this dissertation. This means that we may have another related discipline(s) in literature that is/are related to each problem, but they are not mentioned. Most of the mentioned disciplines are selected according to their usage in the proposed framework.

2.2 Accommodation with digital technologies

The accommodation of tangible CoCs with digital technologies is based on two questions: How the CoC can be represented to accommodate the digital technologies and what are the current works published in literature to represent and improve the forensic information? Regarding the “How”, the web of data vision is discussed to depict how it can be used to represent such information (Section 2.2.1). Regarding the “What”, the CF and digital evidence discipline is provided to explain different forensic processes and all efforts that have been performed to represent and improve such information (Section 2.2.2).

2.2.1 Semantic web and web of data

Semantic web is an extension of the current web (i.e., from document to data) (Berners-Lee et al., 2001; Bizer et al., 2009), designed to represent information in a machine-readable format by introducing a standard model called RDF. RDF is originally designed as a metadata model to model and interchange information on the web (Beckett et McBride, 2004).

The classical way for publishing documents on the web is just by presenting them in HTML (Hyper Text Markup Language)⁴ format, naming these documents using URI, and linking them through hypertext links called hyper text anchors of HTML. These facts allow the consumer to navigate over the information on the web by using a web browser application and crawling over the information by typing keywords in a search engine that is using the support of HTTP. This is called “the web of documents” (see Figure 2.2).

As shown in Figure 2.2, the current web contains different sets of HTML documents, connected to each other through hyper-links. They are consumed using web browsers and search engines.

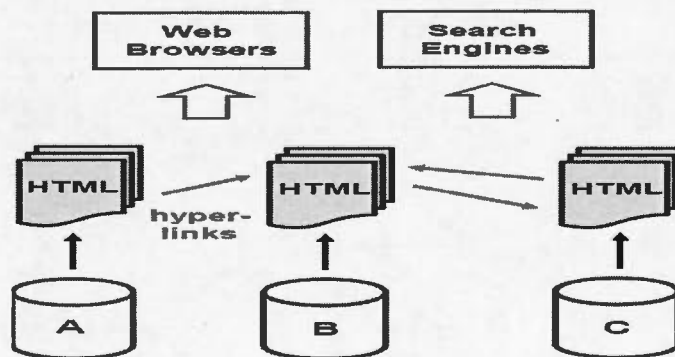


Figure 2.2 Web of documents

⁴ <http://www.w3.org/html/>

With the same analogy, entities and contents (i.e., data) within documents can be linked using typed links and with the same principles used by the web (i.e., web aspects). This is called “the web of data” (see Figure 2.3).

Nowadays, the main aim of the semantic web is to publish data on the web in a standard structure, and in manageable format (Campbell et MacNeill, 2010). Tim Berners-Lee outlined the principles of publishing data on the web. These principles, known as Linked Data Principles (i.e., LD principles) (Berners-Lee, 2006; Bizer et al., 2009; Omitola et al., 2011), are the following:

- Use URI as names for things (Berners-Lee et al., 2004).
- Use HTTP-URIs so that people can look up those names (Fielding, 2014).
- When someone looks up a URI, useful information can be provided using the standards (RDF, SPARQL Protocol and RDF Query Language) (Prud’Hommeaux et Seaborne, 2008), where RDF is a universal data format, and SPARQL is a standard query language for RDF).
- Include RDF statements that link to other URIs so that they can discover related things.

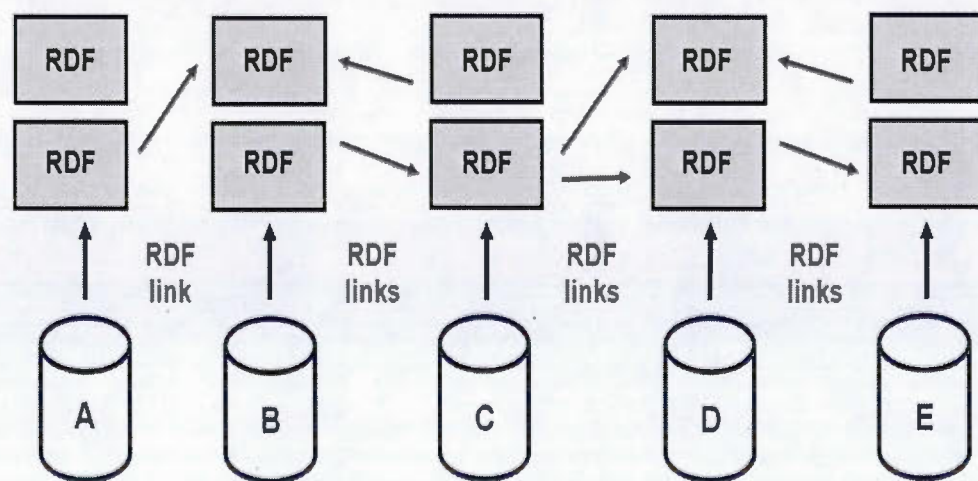


Figure 2.3 Web of data

According to the W3C recommendation, RDF is a foundation for the encoding, exchanging, and reusing of structured metadata (Beckett et McBride, 2004). It can be serialized using different languages (e.g., RDF/XML (Beckett et Berners-Lee, 2011), Turtle (Beckett et Berners-Lee, 2012), RDFa (Adida et al., 2004; W3C, 2015), N-Triples (Beckett, 2014), N3 (Berners-Lee et Connolly, 2011)). RDF consists of three slots. The three slots form a triple: subject-predicate-object or resource-property-value. Next paragraphs illustrate briefly these slots.

In RDF, predicate is the second part of an RDF statement. Unlike object, a predicate and subject must always be a Uniform Resource Identifier (URI). The predicate establishes the relationship between a subject and object and makes the object value a characteristic of the subject (i.e., indicates what kind of relation exists between subjects and object, for example, this is the name or date of birth). The predicate URIs comes from vocabularies, collections of URIs that can be used to represent information about a certain domain.

Subjects and Objects are the first and third part of a statement, respectively. The subject of an RDF statement is either a URI or a blank node (anonymous resources), both of them denote resources. Objects can be resources and values, IRI/URI/URL, literals, or blank nodes (when mentioning resources without global identifier). Literals can be basic values (plain) or IRI (typed). A plain literal is a string combined with an optional language tag (e.g., 4th of July, 5.12, Lacoste).

```
<dc:title>Walking on street</dc:title>
```

With an optional language:

```
<dc:title> <pcv: Descriptor>
```

```
<pcv:label xml:lang="en">Walking on street</pcv:label>
```

```
</pcv: Descriptor> </dc:title>
```

Typed literal is a string combined with a datatype URI that identifies the datatype of the literal (e.g., common datatypes such as integers, dates, floating point defined by XML)⁵.

```
<rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date
rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1999-08-16
    </exterms:creation-date>
</rdf:Description>
```

RDF also identifies things using web identifiers (URIs) and describes resources with properties and property values: resource-property-value. Resources have properties (attributes) that admit certain range of values or that are pointing to other resources. A resource is anything that can have a URI describing an entity from the web (e.g., persons, places, web documents, pictures, etc.). Resources can be meaningfully placed in a class. A class (or classification) is a meaningful way of grouping resources. When any resource is placed into a class, it is called an individual of that class (also sometimes called an instance of the class). For example, a feline class, a class for all members of the feline species:

tutorial:Feline *rdf:type* rdfs:Class

If we place the cat berry to this class Feline, then Berry resource will be an individual (instance) from the class Feline:

thing:berry *rdf:type* tutorial:Feline

⁵ <http://www.infowebml.ws/rdf-owl/Literal.htm>

This formally means that the resource berry (with its unique subject URI) is an individual (or, member) of the class with identifier given by the object URI.

A property in RDF allows us to define or describe characteristics of individual of a class. It is a resource that has a name (e.g., author, homepage) and property value is the value of a property (e.g., 56) or a resource (e.g., <http://www.w3schools.com>).

If the object is a literal, then we will have literal triples and this type will be used to describe the properties of resources. If the object is a resource, then we will have RDF Links and this type will describe the relationship between two resources. In this case, the predicate position defines the type of relationship between resources.

RDF resources were represented by Uniform Resource Identifiers (URIs) of which URLs are a subset. URI is a string of characters used to identify a name or a web resource (Berners-Lee et al. 2014). Recently, URIs have been upgraded to International Resource Identifier (IRI) ⁶. The difference is that the former supports only ASCII encoding (i.e., 1 byte encoding), while the IRI is fully international characters generalizing the URIs and use the UTF-8 encoding (i.e., variable length encoding, 1-4 bytes).

URI and HTTP are the two essential technologies of the web upon which the LD relies. URI can be used to identify and represent any entity that exists in the real world. It identifies a resource either by name, location or both.

All Unified Resource Locators (URLs) and Unified Resource Name (URNs) are URIs, but not all URIs are URLs or URNs. When a URI identifies a resource using name in a given namespace, but doesn't specify how the resource is obtained, then this URI is called a "URN" (e.g., this may appear in XML documents to define a namespace, *targetNamespace="urn:example"*, where a "targetNamespace" uses a

⁶ <https://www.w3.org/International/iri-edit/draft-duerst-iri-04>

URN to define an identifier to the namespace). When a URI identifies a resource using the network location using access mechanism (i.e., HTTP, File Transfer Protocol FTP, etc.), then this URI is called a “URL” (e.g., <http://test.com>, <ftp://test.com>).

On the web, any URI is always accompanied by the HTTP, which makes the entity being represented, dereferenceable/resolvable to more resources. Both technologies were integrated with HTML to structure and link web documents. Nowadays, the data presented in these documents are integrated using the RDF and URI HTTP to structure and link different data and resources.

URIs are meant to identify web document or a real world object using Hash URI or 303 URI⁷. The essential thing to publish data is to have a unique domain/namespace minted by unique URL owned by the publisher. As mentioned, URI HTTP is used to relate and identify real-world objects and abstract concepts and thereby maximizing the discoverability of more data (i.e., resources). Thus, URIs need to be dereferenceable to identify real objects (Sauermann et al., 2011). Objects and documents should not be confused with each other; therefore, a common practice called “content negotiating” is used by an HTTP mechanism (Fielding, 2014). It sends HTTP headers with each request to indicate what kind of documents they prefer. Servers can then inspect these headers and select an appropriate representation of resources: HTML document or RDF document.

In addition, URIs can be used to distinguish between the thing/resource and a web document describing this thing/resource. Two different types of URIs can be used by the content negotiation for non-information resources (Sauermann et al., 2011; Berners-Lee et al., 2014):

⁷ <https://www.w3.org/TR/cooluris/>

- **303 URIs (known as 303 redirect):** when the URI identifying non-information/non-document resource is dereferenced (i.e., called first request, from client to server), the server used redirects the client request to see another URI of a web document (i.e., called second request, from server to client), which describes the concept in question. This redirection is called “303 redirect”. To elaborate on the idea, this redirection occurs when the server can not return a representation of the requested resources. At this time, the server sends back to the client the URI of an information resource describing the non-information resource. URIs related to non-information resource can have three different patterns:
 - URI identifying resource ‘x’ itself:
(e.g., <http://www.example.com/resource/x>)
 - URI identifying the serialized RDF document (i.e., serialized using RDF/XML, Turtle, N3 or any other language) describing the resource ‘x’:
(e.g., <http://www.example.com/data/x.rdf>)
 - URI identifying the HTML document describing resource ‘x’:
(e.g., <http://www.example.com/page/x.html>)
- **Hash URIs:** This type of URI is another way of naming non-information resources to avoid two http requests used by the 303 URIs. Its format contains the base part of the URI and a fragment identifier separated from the base by a hash symbol (e.g., <http://www.example.com/about#x>). When a client requests Hash URI, the fragment part is stripped off by the HTTPS protocol before requesting the URI from the server. This means that the Hash URI does not necessarily identify a web document and can be used to identify real-world objects. If clients strip off the fragment part before requesting a Hash URIs, it results in an absolute URI that identifies a document in which the same thing

has been described using Hash URL. In this case the content negotiation could be employed to redirect the absolute URI (<http://www.example.com/about>) to either an HTML or an RDF representation.

Using the first type of URI, publishers could publish the description of any concepts (e.g., real world object: persons) on their servers using two types of representations: HTML document containing a human-readable representation about a concept 'x', and RDF document about the same concept 'x'. This can be done by publishers using the three different patterns described above (Berrueta et al., 2008; Heath et al., 2008; Heath et Bizer, 2011).

Using the second type of URI, publishers can define different vocabulary terms in order to describe their data that they want to publish on the web. They may also use the Hash URI to serve an RDF/XML file containing the definitions of all these vocabulary terms. After the resources are identified using URIs, they are connected together using different types of RDF links (Heath et Bizer, 2011):

- **Relationship Link:** this type of link relates a resource to different resources in other data sources
- **Identity Link:** this link is used to link two or more URI when they are representing the same real-world object. This type of link is useful to retrieve more information about a resource and map it to other identical resources.
- **Vocabulary Link:** this link is used to link between data instances (i.e., A-Box, Assertion Box) and the definitions of vocabulary terms (T-Box, Terminology Box) are used to represent and publish this data instances. Also, it can link the definitions of two terms together. This elaborates the fact that each term will be self-descriptive and dereferenceable to more resources.

Once these links are provided, they create a global data graph that spans data sources and enable the resolvability between different resources within different data sources.

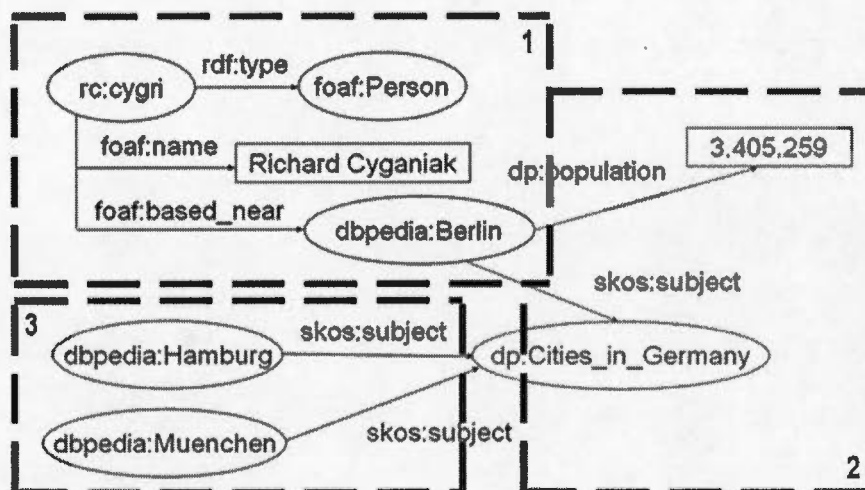


Figure 2.4 RDF Models (Heath et al., 2008)

Figure 2.4 depicts how all these concepts can be realized in a RDF model. In this figure, a person called “*Richard Cyganiak*” identified himself by URI <http://richard.cyganiak.de/foaf.rdf#cygri> (i.e., Hash URI), and he used the “*rdf:type*” (i.e., vocabulary link) to specify the Person class, and the name property imported from FOAF vocabulary to specify his name and location (i.e., relationship and vocabulary link). He stated that he is near Berlin by using the URI <http://dbpedia.org/resource/Berlin> (i.e., 303 URI and relationship link) represented in the name space “*dbpedia:Berlin*”. The latter is dereferenced and can be a subject for another RDF graph describing the City Berlin in more detail including its population and the country in which this city is located (i.e., relationship link).

Finally, the third RDF graph used the name space object of the second graph to provide the other cities that are located in Germany using the SKOS (Simple Knowledge Organization System) (Isaac et Summers, 2008; Pastor et al., 2009) (i.e., relationship link).

Identity links may also exist if one or more resources with different URIs refer to the same real world object/concept. Another resource that may exist in another data source that describes Hamburg is called the “*Hanseatic*” city. Hamburg and Hanseatic city are equivalents in the real world and may be mapped together using the “*owl:sameAs*” constructor (i.e., identity link).

The Linked Open Data (LOD) cloud project is a visible example that uses all these structures and adapts the technologies provided by the web of data to build the LD.

2.2.1.1 Linked Open Data (LOD)

The era of LD started from October 2007, after Tim Berners-Lee underlined explicitly the principles of publishing data on the web. Researchers of the semantic web consider that there was no semantic web before the LOD web project, as there were no explicit guidelines to aid all people to publish their data in a well-linked manner.

However, long before 2007, since 1997, the semantic web was built on URIs and RDF. Data was published following the LDP, but the only difference is that they were not underlined explicitly at that time. Prior to the LDP, and before 2007, there were two ways of publishing RDF on the web:

- The first way was to publish dumps of RDF and OWL and not linking them to other data sources (i.e., no connection using URIs and no links could be followed for navigation). Their concerns were meant only to foster and enrich the semantics by mainly focusing on ontologies; the decidability of the languages and the modeling methodologies. This way was used by most of those people who published data on the web.

- The second way was to publish RDF in a linked-data manner using typed links, and it was used by a minority of people. This group concentrated mainly on web aspects, and linking data using typed links.

LDP is the calling name to what the second set of publishers was doing. It laid out some guidelines (i.e., aspects that were underlined explicitly by Tim Berners-Lee in 2006) and encouraged people to follow them. Thus, LD was not just a switch that was turned on in 2007. It has roots in what the second group of RDF publishers was doing a long time before.

As the major shift in research, the development community surrounding the semantic web has moved from the semantics towards the web aspects. Semantics focus on constructing complete ontologies, while web aspects require fewer axioms by using lightweight ontology and connection through URI and typed links. Thus, the concern shifts from concentrating on the ontologies and their semantics, to focusing on the web aspects (i.e., LDP) and how to publish and consume data on the web.

Today, the Linked Open Data (LOD) project is the most visible project using the technology stack of the web (URLs, HTTP, and RDF) and converts existing open license data into RDF according to the LDP (Berners-Lee, 2006; Campbell et MacNeill, 2010) (see Figure 2.5 for last state of LOD cloud 2014).

As mentioned, the LOD is based on the LDP, where URI resources are linked using typed RDF links to other resources within the same or to other datasets. Two direction links can be used: links to navigate forward and others to navigate backward between resources. For example, if we have an RDF triple connecting the two resources *x* and *y*, and we need to move forward from *x* to *y*, then this RDF triple should appear in the document describing the resource *y*. This triple is then called an incoming link, because it allows to navigate back to resource *x* and it can be thought

of as an *incoming link* to y ⁸. It is the same case for the outgoing link, where the RDF triple should appear in the document describing the resource x and allows it to navigate forward to resource y (Alexander et al., 2009).

Ontologies are then used to foster and serve the semantic interoperability between parts that want to exchange such data. These are known as lightweight ontologies (Hitzler et Harmelen, 2010) that use the full advantages of semantic web technologies, minimize OWL constructs, and reuse existing RDF vocabularies wherever possible.

While RDF provides the model and syntax for describing resources, it does not define the meaning of those resources. That is where other technologies such as RDF Schema (RDFS) come in (W3C, 2014). RDFS specifies extensions to RDF that are used to define the common vocabularies in RDF metadata statement and enables specification of schema knowledge. It develops classes for both resources and properties. However, RDFS is limited to a subclass hierarchy and a property hierarchy with domain and range definitions of these properties. RDFS limitations are range restrictions, incapable of expressing disjointness between classes, the combination between classes, cardinality restriction, and characteristics of properties (W3C, 2004).

Thus, RDF is the standard format to create LD and it is sufficient to use the constructors of RDFS and some features of OWL to represent data in an LD structure. Combination of constructors from both vocabularies (i.e., RDFS and OWL) represents the lightweight ontology of RDF and LD. This is known by RDFS++⁹.

⁸ <http://linkeddatabook.com/editions/1.0/#htoc42>

⁹ <http://linkeddatabook.com/editions/1.0/#htoc50>

Figure 2.5 shows part of the LOD data project cloud diagram, where links exist between items in two connected datasets. Some datasets are connected together using the outgoing links, the incoming links or both.

The next subsections highlight all the RDFS constructors and some OWL primitive constructors that will be used to construct the first two modules of the CF-CoC framework mentioned in Chapter 3.

The RDFS and OWL constructors are classified according to the term type (see Table 2.1 and Table 2.2). A term *X* can be defined as a class (i.e., a class like “*rdfs:Class*”) or as a property (i.e., property like “*owl:ObjectProperty*”). This definition takes place before the term will be used (i.e., before publication, T-Box: terminological box). Later, the defined terms are used to describe and publish various data (A-Box: Assertion Box) (Dean et al., 2004; McGuinness et Van Harmelen, 2004; Van Harmelen et McGuinness, 2004). The type of the term also determines its slot position during publication (i.e., when the term is a property, it always occupies the predicate slot. However, if it is a class, it can be a subject or object).

2.2.1.2 RDF Schema (RDFS) constructors

As mentioned, RDF Schema is the semantic extension of the basic RDF vocabulary. It is used to provide a data-modeling vocabulary for RDF data. The RDFS constructors are used to define terms, which are used to express groups of similar resources (W3C, 2014).

RDFS differs from a classic object-oriented system¹¹. The latter defines a class in terms of the properties its instances should have. However, the former describes

¹¹ <https://www.w3.org/TR/sw-oosd-primer/#comparison>

properties in terms of the classes of resource to which they apply (i.e., domain and range described in the next table, Table 2.1). This fact is beneficial for others to re-define the original description of these classes. For example, RDFS can define a property “*eg:author*” to have a domain of “*eg:Book*” and a range of “*eg:Person*”, while the object-oriented system can define a class “*eg:Book*” with an attribute called “*eg:author*” of type “*eg:Person*”. Thus, in RDFS, it is easy for others to subsequently define additional properties instead of re-defining the original description of these classes. This benefit is considered to be one of the architectural principles of the web (Berners-Lee, 2006; W3C, 2014). Table 2.1 summarizes the constructors of RDFS vocabulary and highlights where such constructors can appear in slots of RDF models. Let us consider that the term in question is named X, which is considered an instance of *rdf:Property* or *rdfs:Class*. The next points elaborate on the constructors of Table 2.1. Let us consider that X and P are properties, Y and C are classes, T is a triple, S is a subject and O is the object in triple T.

- *X rdfs:subPropertyOf P*

Any property denotes a relation between resources. “*rdfs:subPropertyOf*” constructor applies to properties and is interpreted as the subset relation between the relations they denote. The “*rdfs:subPropertyOf*” is an instance of “*rdf:Property*” and states that all resources related by X are also related by P. This means if $T(S, X, O)$, and X is sub-property of P, $T(X, rdfs:subPropertyOf, P)$, and both are instances of “*rdf:Property*”, this implies that T, where T is a constructor, can have also P as predicate $T(S, P, O)$. For example, if “*mother*” is a sub property of “*parent*”, if there is a valid triple (a, mother, b) then the triple (a, parent, b) is also valid.

- *X rdfs: range Y*

States that X is an instance of the class “*rdf:Property*”, Y is an instance of the class “*rdfs:Class*” and that the resources denoted by **the object O** of T (S, X,

O) whose predicate is X are instances of the class Y. For example, if mother has a range Person $T(\text{mother}, \text{rdfs:range}, \text{Person})$, and mother is a predicate in $T(\text{Alice}, \text{mother}, \text{Eve})$, then Eve is the instance of the class Person.

Table 2.1 RDFS Constructors for Property and Class Terms (W3C, 2014)

If X is an instance of <i>rdf:Property</i>	
<i>rdfs:subPropertyOf</i>	Is an instance of <i>rdf:Property</i> that is used to state that all resources related by one property are also related by another. For example, the triple $X \text{ rdfs:subPropertyOf } X'$ states that X is an instance of <i>rdf:Property</i> , X' is an instance of <i>rdf:Property</i> and X is a subproperty of X'. Also the domain and range of <i>rdfs:subPropertyOf</i> is <i>rdf:Property</i>
<i>rdfs:range</i>	Is an instance of <i>rdf:Property</i> that is used to state that the values/resources of a property X are instances of one or more classes. For example, the triple $(X \text{ rdfs:range } C)$ states that X is an instance of the class <i>rdf:Property</i> , that C is an instance of the class <i>rdfs:Class</i> and that the resources denoted by the <u>objects</u> of triples whose predicate is X are instances of the class C
<i>rdfs:domain</i>	Is an instance of <i>rdf:Property</i> that is used to state that any resource that has given property X is an instance of one or more classes. For example, the triple $(X \text{ rdfs:domain } C)$ states that X is an instance of the class <i>rdf:Property</i> , that C is an instance of the class <i>rdfs:Class</i> and the resources denoted by the <u>subjects</u> of triples whose predicate is X are instances of the class C

If X is an instance of <i>rdfs:Class</i>	
<i>rdfs:subClassOf</i>	Is an instance of <i>rdf:Property</i> that is used to state that all the instances of one class are instances of another. For example, the triple (<i>X rdfs:subClassOf X'</i>) states that <i>X</i> is an instance of <i>rdfs:Class</i> , <i>X'</i> is an instance of <i>rdfs:Class</i> and <i>X</i> is a subclass of <i>X'</i>
Common constructors between property instance and class instance	
<i>rdfs:comment</i>	Is an instance of <i>rdf:Property</i> that is used to provide a human-readable description of a resource
<i>rdfs:label</i>	Is an instance of <i>rdf:Property</i> that is used to provide a human-readable version of a resource's name

- *X rdfs:domain Y*

States that *X* is an instance of the class "*rdf:Property*", *Y* is an instance of the class "*rdfs:Class*" and that the resources denoted by **the subject S** of *T(S, X, O)* whose predicate is *X* are instances of the class *Y*. For example, if mother has a domain person *T(mother, rdfs:domain, Person)*, and mother is a predicate in *T(Alice, mother, Eve)*, then Alice is the instance of the class Person.

The simple case is to have a single class for domain and range, respectively. The constructor of *owl:intersectionOf* is the default semantics of multiple classes in domain/range in RDFS. For example, if a Father is an intersection between Parent and Male, this means that a Father is exactly a parent who is also a Male. A Person is union of Female or Male; this means that every person is either Male or Female.

The standard way to create multiple domains and ranges for the same object property is through deciding whether the union “*owl:unionOf*” constructor or the intersection “*owl:intersectionOf*” constructor hold between classes (i.e., both are constructors from OWL vocabulary and both cannot be expressed in RDFS). Both have different meanings, and which one to use depends on what we want to express in our ontology. For example, let us say you have a property P and its domain/range is defined using two classes on the triples $T_1(P, \text{rdfs:domain/range}, A)$ and $T_2(P, \text{rdfs:domain/range}, B)$. If we use the union, this means that any individual that is the subject/ (object) of a property P must be an instance of A **or** B. However, if we use the intersection this means that individual that it is the subject/ (object) of a property P must be an instance of A **and** B (i.e., see below for an example that is serialized and expressed using turtle language):

In cases where we need to make a disjunction (union) of the classes A and B, and “a” is an individual, then

: myProperty *rdfs:domain* [a owl: Class; *owl:unionOf*(:A :B)]

In cases where we need to make a conjunction (intersection) of the classes A and B, and “a” is an individual then

: myProperty *rdfs:range* [a owl: Class; *owl:intersectionOf*(:A :B)]

- *X rdfs:subClassOf C*

Classes are resources denoting a set of resources, by the meaning of the property “*rdf:type*”. The constructor “*rdfs:subClassOf*” is an instance of “*rdf:Property*” that is used to state that all instances of one class are instances of another. This means if $T(S, \text{rdf:type}, X)$, and X is a subclass of C $T(X, \text{rdfs:subClassOf}, C)$ and both are instances of “*rdf:Class*”, it is implied that S is of type C $T(S, \text{rdf:type}, C)$. For example, if Alice is of type woman $T(\text{Alice}, \text{rdf:type}, \text{Woman})$, and woman is a subclass of class person $T(\text{Woman},$

rdfs:subClassOf, Person), where woman and person are instances of “*rdf:Class*”, then this implies that Alice is also a person $T(\text{Alice}, \text{rdf:type}, \text{Person})$.

- *X rdfs:label L*

The constructor “*rdfs:label*” is an instance of “*rdf:Property*” that is used to provide human-readable version of a resource’s name. For example, if we have a short name for a resource R, the label can provide a human-readable word or phrase describing R. L, here, is of type *Literal*. This means that the range is “*rdfs:Literal*” and its domain is the class *rdfs:Resource* itself, $T(R, \text{rdfs:label}, L)$.

- *X rdfs:comment L*

The constructor “*rdfs:comment*” is also an instance of “*rdf:Property*” that is used to provide human-readable description of a resource R. It is the same idea of label constructor; its range is the “*rdfs:Literal*” and its domain is the resource itself, $T(R, \text{rdfs:comment}, L)$.

2.2.1.3 Web Ontology Language (OWL) constructors

Web Ontology Language is richer with additional vocabulary than that supported by RDFS to add more restrictions to the knowledge representation by defining objects and their relationship and adding restrictions on properties. For example, relationships between classes (e.g., *disjointWith*), equality (e.g., *sameAs*), richer properties (e.g., *symmetrical*) and class property restrictions (e.g., *allValuesFrom*). Thus, these extra features answer the limitation of the RDFS.

In LD, OWL constructors are not fully deployed. Only few constructors are mainly used to map between property and class terms. Other constructors are used to relate

and build relationships between various properties. The primitives selected from the OWL for LD are provided in the next table (see Table 2.2) (Dean et al., 2004).

Table 2.2 OWL Constructors for Property and Class Terms (W3C, 2014)

If X is a term of type (<i>rdf</i> : type) Property (<i>rdf</i> : Property / <i>owl</i> : ObjectProperty)	
<i>owl</i> : <i>equivalentProperty</i>	This constructor is used to map between two terms of type <i>Property</i>
<i>owl</i> : <i>inverseOf</i>	This constructor is used to state that one property is the inverse of another. It is used to describe inverse relation between properties.
<i>owl</i> : <i>InverseFunctionalProperty</i>	Whenever X property is used as a predicate in a triple, its object will have one and only one subject. Thus, each object should be able to uniquely identify a subject. This constructor is a sub class of <i>owl</i> : <i>objectProperty</i>
<i>owl</i> : <i>FunctionalProperty</i>	Same idea as the last constructor, but here, when X is defined to be of type <i>FunctionalProperty</i> , each subject, where X is a predicate, can have at most one object. This constructor is a subclass of <i>rdf</i> : <i>Property</i>
If X is a term of type (<i>rdf</i> : type) Class (<i>rdfs</i> : Class)	
<i>owl</i> : <i>equivalentClass</i>	This constructor is used to map between two terms of type Class
Common Constructors between Property and Class terms	

<i>owl:sameAs</i>	Two URI terms can be mapped together using the <i>sameAs</i> constructor to refer to the same resource
-------------------	--

The next points elaborate on the constructors provided in Table 2.2. Let us consider that X and Y are properties of type “*rdf:Property*”:

- *X owl:inverseOf Y*

Simply, as it is shown in this structure, X is the inverse of Y if we read it from left to right. At the same time, we can deduce that Y is the inverse of X from the other direction (i.e., from right to left). In practice, it is also useful to define relations between properties in both directions. This relation is exactly the same as the passive voice in grammar. If we have an axiom of the form P1 *owl:inverseOf* P2 asserts that for every (x,y) in the property extension of P1, there is a pair (y,x) in the property extension of P2, and vice versa.

For example, “people own cars” means the same thing as “cars are owned by people”. This means that when “*owl:inverseOf*” is used between two properties in a triple, the domain of a property is the range of the other and vice versa. Thus, “*owl:inverseOf*” is a symmetric property.

The *inverseOf* property is also useful to work on individual resources, for example it is used for inverse roles (e.g., *isChildOf* \equiv *hasChild*).

- *X as owl:InverseFunctionalProperty*

In this structure, when a property X is tagged as “*InverseFunctionalProperty*” in T(S,X,O), then the object O of a property X statement uniquely determines the subject S (some individual) . Further when another subject S’ is linked to the same object O through predicate X, then the S’ is actually the same subject S (i.e., S’ *owl:sameAs* S). Thus, in this case, the object O uniquely

determines the same individual subjects (i.e., S' and S are two different names for the same thing).

For example, the value of the property Social Security Number (SSN) is assigned to one, and only one, person. To represent this information in its correct semantic, SN should be tagged as *"InverseFunctionalProperty"*, where its domain will be of type person (*"foaf:Person"* of type *"rdf:Class"*), and its range will be a literal (*"rdfs:Literal"*) (e.g., T(Peter, SN, 306305)). Then any mentioned subjects (e.g., Peter, *myBrother*, *myCousin*, etc.) published by any publisher refer to the same person.

- *X as owl:FunctionalProperty*

In this structure, when a property X is tagged as *"FunctionalProperty"* in T(S,X,O), then X is a property that can have only one (unique) value for each instance (i.e., resource) S. the values of O cannot have two distinct values (at most one value). For example, any woman can have either one, and only one, husband which is a man or no husband at all (i.e., this example is culture dependent and may change from one country to another. It is used under the assumption that each woman may have only one husband). So, to express this information semantically correct, *"hasHusband"* in T (Alice, *hasHusband*, Bob) is a property of type *"FunctionalProperty"*, where its domain and range are of type person (i.e., *"foaf:Person"*).

- *X owl:equivalentProperty Y*

The main aim for this constructor is to map between properties from two ontologies and relate the same subject resource to the same value object resource where both are properties. For example, if we have (X, *rdfs:subProperty*, Y) and (Y, *rdfs:subProperty*, X) \Leftrightarrow (X, *owl:equivalentProperty*, Y). This fact saves much effort in developing

ontology in ways to have simple and useful implications and facilitate the task for an OWL reasoners to derive a value for some resource's X if it can find a value for resource Y (Dean et al., 2004; W3C, 2004), because the *owl:equivalentProperty* hold between two properties that have the same “values” (i.e., same property extension), but both of them have different intentional meaning (i.e., denote different concept)¹²:

```
<owl:ObjectProperty rdf:ID="lecturesIn">
    <owl:equivalentProperty rdf:resource="#teaches"/>
</owl:ObjectProperty>
```

In $T(X, owl:equivalentProperty, Y)$, X is a property in the left slot, and Y is another property from the right slot. We can deduce that the domain and range of the “*owl:equivalentProperty*” constructor are the same (*rdf:Property*). The “*owl:equivalentProperty*” is intended for RDFS/OWL properties.

For example, if we have two different ontologies (namespaces): ns and DC (DC, 2015), both of them have a property called “title”, and they are equivalent with the same values $T(dc:title, owl:equivalentProperty, ns:title)$ and we have another triple $T'(book, dc:title, roman)$, where the book is a resource and roman is of type Literal, then we can infer : $T_{inferred}(book, ns:title, roman)$, where *ns:title* is also of type *rdf:Property*.

- *X owl:equivalentClass Y*

This property holds the same explanation mentioned in the point of “*owl:equivalentProperty*”, but the difference is the mapping between two classes instead of two properties. So, X and Y are of type “*rdfs:Class*”. By using this constructor, two class descriptions involved have the same class

¹² <http://www.infowebml.ws/rdf-owl/equivalentProperty.htm>

extension (i.e., have exactly the same set of individuals) (Dean et al., 2004). Also, as mentioned before, “*equivalentProperty*” does not imply property equality. This is also the same case with “*equivalentClass*” (i.e., equivalence between two classes means both class extensions contain exactly the same set of individuals). The “*equivalentClass*” does not imply class equality (i.e., class equality means that classes denote the same concept). However, “*owl:sameAs*” is the constructor that can be used to treat equality between classes.

The “*owl:equivalentClass*” constructor also differs from the “*rdfs:subClassOf*”. In subclass, the relationship is hierarchical, and in one way, direction from child class to parent class. Thus, if A is a subclass of B, this restricts A to necessarily inherit all characteristics (properties) of B. Further, all instances of A must necessarily have all properties of B, but the “*owl:equivalentClass*” can go in both directions between both of them. This means, if A and B are two equivalent classes, then A *rdfs:subClassOf* B in one direction and B *rdfs:subClassOf* A in the other direction (see Section 2.2.1.4 for its entailment rule).

- X *owl:sameAs* Y

In contrary with the *equivalentClass* and *equivalentProperty*, the *sameAs* is used to define equality by stating that two URI references actually refer to the same individual/thing. This means that both URIs denote the same concept. Links and individual to an individual occurs when both individuals have the same identity. For example:

```
<rdf:Description rdf:about="#Jean_Claude_VanDam">
  <owl:sameAs rdf:resource=VanDam"/>
</rdf:Description>
```


The *sameAs* constructor is used to state that seemingly two different individuals (e.g., Jean_Claude_VanDam and VanDam) are actually the same person.

However, the first former constructor - *equivalentClass* is used to state that classes are extensionally equivalent (i.e., have exactly the same sets of members/individuals), but it does not imply class equality, for example:

```
<footballTeam owl:equivalentClass us:soccerTeam/>
```

This example states that the two classes have the same class extension. This means that both classes have exactly the same set of individuals, but are not necessarily the same concept. They are not equated but are equivalent.

When the “*owl:sameAs*” constructor is used to relate the same classes/properties, this means that the two classes/properties are to be interpreted as the same object/individual. This occurs only in OWL FULL, where a class can be treated as instances of (meta) classes and thus pushes the ontology out of OWL DL. Using again the same example, we can use the *owl:sameAs* to define equality between FootballTeam and SoccerTeam, thus indicating that the two classes have the same intentional meaning :

```
<owl:Class rdf:ID="FootballTeam">
  <owl:sameAs rdf:resource:http://sports.org/US#SoccerTeam>
</owl:Class>
```

For the second former constructor - *equivalentProperty* is used to mention that two properties are equivalent not equated and when we want to equate them we can use the *owl:sameAs* in the OWL Full.

The “*sameAs*”, “*equivalentProperty*”, “*equivalentclass*” are three constructors that relate terms intensionally and extensionally. The constructor “*sameAs*” is used not only to equate between two individuals, but also between classes and properties. This

means that classes and properties can be also treated as individuals, and this only valid in OWL Full. OWL Full allows free mixing of OWL with RDF Schema and, like RDF Schema, does not enforce a strict separation of classes, properties, individuals and data values. When “*sameAs*” equates individuals, both individuals will have same intentional meaning because both individuals denote the same concept.

The other two constructors, “*equivalentProperty*”, “*equivalentclass*”, can be used to state that two terms (property/class) have the same extensions. For classes: two classes are equivalent when both of them have exactly the same set of individuals. For properties: two properties are equivalent when both of them have the same values. This means that these terms will have same extensional meaning (not same concepts).

All constructors provided in Tables 2.1 and 2.2 are enough to define and publish data on the web. Publication of terms on the web passes through three steps: identification, definition and publication.

Identification of terms is about how to identify and select terms describing the domain of interest (i.e., these terms are the entities whose properties and relationships can be used later in the publication of data). This step is achieved through the descriptions of different processes and tasks performed within the domain of interest. The identified terms are also called custom or proprietary terms.

In the second step, the identified terms are then defined using different constructors of RDFS (W3C, 2014) and OWL (Dean et al., 2004; McGuinness et Van Harmelen, 2004; W3C, 2004), and uniquely named by HTTP URIs.

In the third step, once terms are identified and defined, they are then published on standardized contents formats. This format is the RDF that provides a generic data model composed of a set of triples where the custom terms occupy one or more slot(s) (i.e., subject, predicate, or object) in these triples. Also, the vocabularies of the

semantic web are used together with the proprietary terms to describe and represent the forensic information.

Before discussing the concepts of defining proprietary terms, the next section presents and underlines some entailments rules that were provided implicitly in the explanation of each constructor.

2.2.1.4 Reasoning on RDFS++

Reasoning depends mainly on the semantic level of the representation of RDFS and OWL that implies a given mathematical formalization for the knowledge base. As mentioned in Section 2.2.1.1, lightweight ontology of LD is a combination of RDFS constructors and some primitives of OWL. These constructors contain a set of inference rules. Inference is a derivation of logical conclusions from premises known or assumed to be true. In RDF, inferences correspond to entailments that derive new assertions from existing ones.

Reasoning is a process to extract new information from existing information stored in a knowledge base. For the LD, the knowledge base is the store where the information is presented in the form of RDF triples. The extracting information process is not limited to extracting or querying triples that are physically stored in the knowledge base. It also infers implicit (i.e., not been explicitly stated) information from these triples.

RDFS and OWL contain a set of inference rules related to their constructors. This section discusses the rules of RDFS constructors, and some rules of OWL (i.e., those that are primitives and used to describe the LD). Table 2.3 depicts the rules of the most used constructors of both vocabularies (i.e., RDFS, and OWL).

OWL Reasoners can be used to reason on RDFS constructors. However, the inverse is not valid, because RDFS is a subset of OWL.

Other entailments rules that can be provided and that are related to the above constructors are:

- *owl:equivalentProperty* :

$$(p1, owl:equivalentProperty, p2), (a, p1, b) \Rightarrow (a, p2, b)$$

$$(a, p2, b) \Rightarrow (a, p1, b)$$

- *owl:equivalentClass* :

If $(c1, owl:equivalentClass, c2)$ and it is associated with another triple $(a,$

$$rdf:type, c1) \Rightarrow (a, rdf:type, c2)$$

$$(a, rdf:type, c2) \Rightarrow (a, rdf:type, c1)$$

But, if the triple of “*owl:equivalentClass*” is not associated with another triple

$$(c1, owl:equivalentClass, c2) \Rightarrow (c1, rdfs:subClassOf, c2), (c2, rdfs:subClassOf, c1)$$

Table 2.3 Rules and entailments of RDFS and OWL¹³

Constructor Name	Rules and Entailments
<i>rdfs:subClassOf</i>	<p><i>subClassOf</i> is a transitive when:</p> $(A, rdfs:subClassOf, B), (B, rdfs:subClassOf, C) \Rightarrow (A, rdfs:subClassOf, C)$ <p>Another Entailment rules of <i>subClassOf</i>:</p> $(a, rdf:type, A), (A, rdfs:subClassOf, B) \Rightarrow (a, rdf:type, B)$ $(A, rdfs:subClassOf, B), (B, rdfs:subClassOf, A) \Rightarrow (A, owl:equivalentClass, B)$
<i>rdfs:subPropertyOf</i>	<p><i>subPropertyOf</i> is transitive when:</p> $(a, p, b), (p, rdfs:subPropertyOf, q) \Rightarrow (a, q, b)$
<i>rdfs:domain</i>	$(p, rdfs:domain, A), (a, p, x) \Rightarrow (a, rdf:type, A)$
<i>rdf:range</i>	$(p, rdfs:range, A), (x, p, a) \Rightarrow (a, rdf:type, A)$
<i>owl:FunctionalProperty</i>	If a property <i>p</i> is tagged as <i>FunctionalProperty</i> then all <i>x, y</i> , and <i>z</i> : $p(x, y)$ and $p(x, z) \Rightarrow y = z$
<i>owl:InverseFunctionalProperty</i>	If a property <i>p</i> is tagged as <i>InverseFunctionalProperty</i> then all <i>x, y</i> and <i>z</i> : $p(y, x)$ and $p(z, x) \Rightarrow y = z$

¹³ <http://semanticweb.org/OWLLD/>

<i>owl:inverseof</i>	If a property p_1 , is tagged as the <i>owl:inverseof</i> p_2 , then for all x and y : $p_1(x,y)$ iff $p_2(y,x)$
----------------------	--

Also, for the constructor “*owl:sameAs*”, there exist several rules, between subjects, objects, predicates.

All the constructors mentioned above represent the relational primitives between classes and properties. Those constructors are used by publishers to define and retrieve implicit information from a triple store (RDF graphs) through RDFS reasoners and SPARQL (Prud’Hommeaux et Seaborne, 2008). Those constructors are the means to reach a lightweight version of semantic web and limit use of ontologies and knowledge representation in order to avoid unexpected inferences when the data are consumed (i.e., in light ontology no much ontological axioms are used, only some little primitives from OWL).

2.2.1.5 Defining proprietary terms

Sometimes there will be cases where new terms need to be developed to describe some aspects of a particular data set. Other times the existing vocabularies are not adequate to describe a particular data set and this is the case of Cyber Forensics, where it is rare to find forensic terms or well-known vocabularies describing it because this domain is still in its infancy and development. Thus, new proprietary terms need to be defined and developed in a dedicated vocabulary, applying the features of RDFS (W3C, 2014) and OWL (W3C, 2004) to describe this particular dataset. However, before creating a new custom term, some aspects (criteria) should be taken into consideration. Some receipts have been provided in (Heath et Bizer, 2011): Search for terms from widely used vocabularies that could be reused to

describe the domain in interest. If the widely deployed vocabularies do not provide the required terms to describe such domains, new terms should be defined as proprietary terms.

- When defining a new term, a namespace owned and controlled by the publisher is required (i.e., unique namespace), in order to mint the new terms to this domain/namespace.
- When creating new terms, a map should be established between these terms and those that are from other existing vocabularies.
- Apply the LDP to the new terms by using the web technology stack (HTTP, URL, and RDF) and this task takes place during the publication process starting from the identification of terms until their publication.
- Label and comment each created term.
- If the term is of type property (i.e., predicate), the domain and range of this term should be determined using the constructors of RDFS and not overloading this new term with ontological axioms.
- If at a later time and after creating a new term, another term was found and enough to be used, an RDF link should be set between the newly created term and the existing one.

Though there exist different guides to publish terms, the process of selecting and identifying them remains a subjective task and depends on the term creator (i.e., we may have two creators selecting and identifying two different terms describing the same concept in the real world). This does not affect the quality of terms being published, because the LDP on the web of data make them self-descriptive. The latter advantage is due to two reasons:

- LDP with naming using HTTP/URIs, offer a dereferenceable nature to the term, so that any LD consuming applications can look up the RDFS/OWL

definitions and retrieve more information about said term – this means that every vocabulary term links to its own definition (Berrueta et al., 2008).

- Publishing mappings between terms from different vocabularies in the form of RDF links (Mendelsohn, 2008).

A related work published in (Brinson et al., 2006) to define an ontology in CF, where an ontological model was created for outlining CF tracks in the education process. This ontology is an ontology with small ‘o’. The small ‘o’ describes situation where classification schemes are being built and refers to the semantic web ontologies in computer science, while capital ‘O’ is a term borrowed from philosophy and it is referring systematic recording of existence and in software system something that exists is something that can be represented by the ontology with small ‘o’ (Poli et al., 2010). This related work discussed how to construct a hierarchical structure for classification of certification domains.

As mentioned, CF is a domain that requires the definition of new proprietary terms. More ontologies with small ‘o’ need to be created. The proposed CF-CoC framework provided in this dissertation will aid the role player to represent CoC by defining new proprietary terms and publish such information on the web of data in RDF format.

Today, the semantic web is made up of linked data. This means that the semantic web is the “what: what we need to achieve” and the linked data is the “how: how we can achieve a semantic web”. Despite this crucial role of linked data, there is no work provided in literature to represent CF information using this technology or representing such information in a lightweight ontology. All the work from the literature try to represent CF information using deep ontologies or using different representation models (state of the art related to representing forensic information will be discussed in Section 2.2.2). This dissertation will discuss (in Chapter 3) the advantages of using LDP to represent CF information.

2.2.2 Cyber forensics and digital evidence

The second discipline in the state of the art section is related to CF and digital evidence. Despite the infancy of the CF field, many works have been provided related to the forensic processes, CoC, and forensic formats.

As mentioned in Section 1.1, the most basic level of a forensic process contains three phases. However, there exist numerous forensic models in literature, each of which relies upon reaching a consensus about how to describe digital forensics and evidence (Andrew, 2007; Köhn et al., 2008). Many works were provided either to explain or to compare between such models.

2.2.2.1 Forensic Processes

The works provided under this category concentrated on the creation of different forensics processes. Different Digital Forensics Process Models (DFPM) have been proposed since 2000 (e.g., Kruse (Kruse II et Heiser, 2001), the United State Department of Justice (USDOJ) (Ballou, 2010), Casey (Casey et al., 2014), Digital Forensics Research Workshop (DFRW) (Palmer, 2001), and Ciarhuin in (Ciardhuáin, 2004)) to assist the players of investigations to reach conclusions upon completion.

A forensic process contains a set of forensic phases that are executed in sequence. Technicians of each forensic process are responsible for providing all forensic information resulting from their investigation, and so on, until the end of a forensic process. Figure 2.6 shows an activity diagram of a forensic process called the Kruse model. Its phases are also mentioned in Figure 1.1.

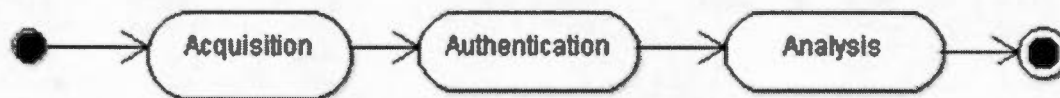


Figure 2.6 Activity diagram of Kruse model

Investigation models are numerous. Many works were provided (Ciardhuáin, 2004; Garfinkel et al., 2006; Köhn et al., 2008; Ballou, 2010; Yusoff et al., 2011; Casey et al., 2014) (see Table 2.4) to explain and compare such models. Some phases from different forensics models may have identical technical requirements, but they differ only in their names (Carrier, 2006). The work presented in (Yusoff et al., 2011) underlines 46 phases from 15 selected investigation models that have been produced throughout 1995 to 2010, and then identifies the commonly shared processes between these models.

Some phases of a forensic model may overlap with another model. For example, the analysis phase is common between USDOJ, DFRWS, and Reith models. Each of those phases is assigned to one or more technicians. Thus, the number of forensic phases and how many technicians are assigned to each phase determine the total number of technicians participating in the forensic investigation. Each forensic phase also contains a set of forensic tasks. In addition, these tasks may overlap and be similar to other tasks in other forensic models.

Table 2.4 Digital Forensics Process Models (Köhn et al., 2008)

	Acquire	Authenticate	Analyze	Collection	Examination	Reporting	Recognition	Identification	Individualisation	Reconstruction	Preservation	Classification	Presentation	Decision	Preparation	Approach Strategy	Returning Evidence	Awareness	Authorization	Planning	Notification	Transposition	Storage	Hypothesis	Proof of defence	Dissemination
Kruse	*																									
USDOJ			*	*	*	*				*	*	*														
Casey							*																			
DFRWS			*	*	*			*			*		*	*												
Reith			*	*	*			*			*	*	*		*	*	*	*	*	*	*	*	*	*	*	
Ciardhuain				*	*								*					*	*	*	*	*	*	*	*	

This section explains the phases of the Kruse model (Kruse II et Heiser, 2001) in detail, since it is the model that encompasses three basic phases of any forensic investigation. The three phases are acquisition of the evidence, authentication of the recovered evidence and analysis of the evidence. The next figure shows the use case diagram of the Kruse model. It shows the three phases of this model and the technician assigned to each phase.

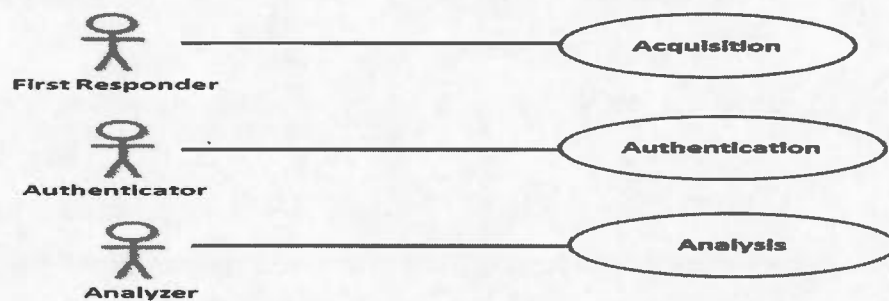


Figure 2.7 Use case diagram of Kruse model

The next three paragraphs briefly explain each phase separately. Also, see Figure 1.1:

- **Acquisition:** this phase is about acquiring digital evidence from digital suspected devices (e.g., closed-scale devices, large-scale devices, etc.). It contains three forensics tasks: state preservation, recovering, and copying. The technician of this phase is called the “first responder” (Köhn et al., 2008; Ballou, 2010; Yusoff et al., 2011).
 - State preservation: the first task is saving the state of the digital device under question, by seizing the machine containing the suspected storage device.
 - Recovery: after seizing the suspected device, the technician tries to recover all deleted files on the device because the main objective of recovery task is to restore the deleted file, especially the system files that record valuable details about this suspected device.

- Copy: after recovering the deleted files, the first responder takes copy from the suspected device to avoid tampering and alteration.
- **Authentication:** the technician of this phase is called the “authenticator”. It is the process of ensuring that the acquired evidence has not been altered and kept its integrity, from the time it was extracted to the time it was transmitted and stored by an authorized source (Menezes et al., 1996). Until normal circumstances, a change to the evidence will render the evidence inadmissible in court. Investigators authenticate the digital media by generating a checksum (Hash) of its contents (i.e., using the MD5, SHA, and CRC algorithms). Checksum is like an electronic fingerprint (i.e., unique numerical value) in that it is almost impossible for two digital media with different data to have the same checksums. The main aim behind this task is showing that the checksums of the seized media (suspected) and the trusted (image) are identical and allows the authenticator to effectively and confidently stand by the integrity of the data in court.
- **Analysis:** this is the last and most time-consuming step in this model. The technician of this phase is the analyzer. In this phase, the investigator tries to uncover the wrongdoing of the crime by examining the acquired data, such as files and directories, in order to identify pieces of evidence and determine their significance and probative value, drawing conclusions based on the evidence found. In (Carrier, 2003), three major categories of evidence are defined that should be considered in the analysis phase:
 - Inculpatory evidence: evidence that supports a given theory (illegal pictures on the hard drive).

- Exculpatory evidence: evidence that contradicts a given theory (time stamp proves that the suspect did not commit the crime).
- Evidence of tampering: evidence which cannot be related to any theory, but shows that the system was tampered with to avoid identification.

2.2.2.2 Improving CoCs

Several works are provided in the literature to improve the CoC. The work presented in (Giova, 2011) provides the idea of exploiting RDF structures to improve an expansible open format of AFF4. In (Cosic et Baca, 2010b), a conceptual Digital Evidence Management Framework (DEMF) was proposed to implement secure and reliable digital evidence CoC. This framework answered the ‘who’, ‘what’, ‘why’, ‘when’, ‘where’ and ‘how’ questions. The ‘what’ is answered using a fingerprint of evidence. The ‘how’ is answered using the hash similarity to changes control. The ‘who’ is answered using the biometric identification and authentication for digital signing. The ‘when’ is answered using the automatic and trusted time stamping. Finally, the ‘where’ is answered using two tracking technologies such as Global Positioning System (GPS) and Radio Frequency Identification (RFID), for geo-location.

Another work in (Cosic et Baca 2010a) discusses the integrity of CoC through the adaptation of hashing algorithm for signing digital evidence by taking into consideration identity, date and time of access of a digital evidence. The authors proposed a valid time stamping provided by a secure third party to sign digital evidence in all stages of the investigation process.

Other published work to improve the CoC is based on a hardware solution. SYPRUS Company provides the Hydra PC solution. It is an entire securely protected, self-contained, portable PC device that is connected to Universal Serial Bus (USB) port, which provides high-assurance cryptographic products to protect the confidentiality, integrity, and non-repudiation of a digital evidence with highest-strength cryptographic technology (Jueneman et LaPedis, 2011). This solution is considered an indirect improve to the CoC, as it protects the digital evidence from modification and violation (Brown, 2009).

2.2.2.3 Knowledge representation of CF processes and CoCs

The work on the knowledge representation created in CF concentrates on the representation of CF models or on digital evidence (as indirect improvement for the CoCs).

An attempt made to represent the knowledge discovered during the identification and analysis phase of the investigation process (Bogen et Dampier, 2004). This attempt uses the Universal Modeling Language (UML) for representing knowledge. It has been extended to a unified modeling methodology framework (UMMF) to describe and think about planning, performing and documenting forensics tasks.

Another work presented in (Köhn et al., 2008) explains how different CF processes are modeled using UML. In this work, the use cases and activity diagrams are presented in order to clarify the limitations of such processes.

Research is also provided in (Schatz, 2007) that proposes that the formal representational approach will be beneficial for the CF. This work summarized the nature of digital evidence and digital investigation at a fundamental level.

Other works are also presented in (Schatz et al., 2004a, 2004b). Indirectly they try to improve the CoC through the representation of digital evidence. Both works concentrated mainly on the representation and correlation of the digital evidence and as an indirect consequence to (im)prove of the CoC.

Recently, a new work is provided in (Al-Fedaghi et Al-Babtain, 2012) to model the forensic process. This work proposed an abstract model for the digital forensic based on the flow-based specification methodology. This methodology is generally used to represent several items, such as data, information, or signals using the Flowthing Model (FM), which contains six stages (arrived, accepted, processed, released, created, and transferred) allowing anyone to draw the system using flow systems.

2.2.2.4 Forensic Formats

Over the last few years, different forensic formats were provided. In 2006, Digital Forensics Research Workshop (DRWS) formed a working group called Common Digital Evidence Storage Format (CDEF) for storing digital evidence and associated metadata (Common Digital Storage Format (CDEF), 2009) surveyed the following disk image main formats (Simson et al., 2006): AFF, Encase Expert Witness Format (EWF), Digital Evidence Bag (DEB), gzzip, ProDiscover, and SMART (now sold under the name of EnCase).

Most of these formats can store limited amounts of metadata, such as case name, evidence name, authenticator name, date, place, and hash code to assure data integrity (CDESF, 2009). The most commonly used formats are described here. AFF is defined by Garfinkel et al. in (Garfinkel et al., 2006) as a disk image container, which supports storing arbitrary metadata, such as sector size and device serial number, in a single archive. The EWF format is produced by EnCase's imaging tools. It contains

checksums, a hash for verifying the integrity of the contained image, and error information describing bad sectors on the source media.

Later, Tuner's digital evidence bags (DEB) proposed a container for digital evidence scene artifacts, metadata, information integrity, and access and usage audit records (Turner, 2005). However, such format is limited to name/value pairs and makes no provision for attaching semantics to the name. It attempts to replicate key features of physical evidence bags, which are used for traditional evidence capturing.

The work in (Cohen et al., 2009) observed problems to be corrected in the first version of AFF. They released the AFF4 user specific metadata functionalities. They described the use of distributed evidence management systems AFF4 based on an imaginary company that has offices in two different countries. AFF4 extends the AFF to support multiple data sources, logical evidence, and several others enhancements such as the support of forensic workflow and the storing of arbitrary metadata. Said work explained that the RDF (Beckett et McBride, 2004) resources can be exploited with AFF4 in order to improve the forensics process model. The authors in this work provided and implemented an architecture that is capable of storing multiple heterogeneous data type that might arise in all modern investigation.

The technician can use any one of these forensic formats. Each forensic tool can generate one or more forensic format(s) that can describe specific forensic results (e.g., AFF4 can be generated by the EnCase imaging tool and provide information about the size of digital media, its chunk size, its chunks in segment, etc.). The technician player is able to manipulate such formats and record different information in his CoC. The framework proposed in this thesis will let the technician to define his own custom terms to describe different forensic information recorded in the CoCs.

The AFF4 is an evolution in forensic imaging technology. The oldest forms of forensic images had several limitations (e.g., raw image or sometimes named *dd*

image, stands for data description). Some of these shortcomings were: the size of the image is exactly the same as the size of the source device (no compression), and the oldest images were not able to keep metadata with image. Metadata must be kept externally and manually associated with the image. AFF4 was able to overcome such limitations by offered compression and started storing metadata within the image. AFF4 uses metadata as its central abstraction by storing all known information in an RDF model (subject-predicate-object). The subject is the globally unique name (URN) generated by GUID (Global Unique Identifier) for an AFF4 object, the predicate is a verb from a known lexicon. All AFF4 statements are stored in a resolver, which is a central point which manages the AFF4 information model. The following is an example serialized in turtle language of URN globally unique identifier for a file found on the path /test/image¹⁴:

```
<aff4://123-abc/test/image>
    aff4:chunk_size 74628;
    aff4:compression <https://www.ietf.org/rfc/rfc1951.txt>;
    aff4:size 2719;
    aff4: stored <aff4://235-abcd>;
a aff4:image
```

The above turtle code provides some information about the image in form of RDF model(e.g., chunk, compression, size), the type of the object URN(e.g., 1230-abc) is AFF4 image, and the object is stored in a volume URN, (e.g., aff4://235-abcd). Integration can be performed between this RDF model and other models.

Briefly, this section (Section 2.2) discussed the works related to the new epoch of the semantic web and different forensic representation models. This dissertation will

¹⁴ <http://www.aff4.org/docs/Overview/Introduction.html>

discuss how to represent and improve the forensic information using the LDP, as well as how the RDFS constructors and some primitives of OWL can describe and define the CoC documents related to the CF domain in order to be published and consumed by the role players and judge, to facilitate the consumption of digital evidence.

2.3 Fostering trustworthiness among role players and judges

Firstly, from the side of the legal system, before digital evidence can be presented for persuasive use, it must be admitted by the judge in court (Insa 2007; Krotoski et al., 2011; Finklea et Theohary, 2012). Admissibility refers to the requirements for evidence to be entered into a court case. There are three common factors that make evidence admitted to the court:

- **Authenticity:** it refers to whether or not the evidence is authentic, or what it is purported to be. This means a process for establishing that digital evidence is what it is represented to be. Authentication refers to legal concepts that promote the integrity of investigation process by ensuring tendered evidence establish what are offered to prove. For example, is the hard drive being seized the correct hard drive that contains the suspected evidence or it has been altered?, therefore some degree of authentication is required.
- **Relevancy:** it refers to the relevance of presented evidence to the case in question. Are the used and provided evidence related to the case, and do they add weighted/significant value to the investigation?
- **Reliability:** refers to whether or not the evidence meets some “minimum standard of trustworthiness”, this means the creditability of a source that is being used as evidence. It is realized through respecting some legal concepts

such as the “*Daubert*” standard or Frye test, which was superseded by Federal Rules of Evidence (FRE) as the standard for admissibility of expert evidence in federal court (Bernstein, 2001). Each country and province may have its own rules. For example, some states in the United States use the “*Daubert*” guidelines (Farrell, 1993). When technical or scientific evidence is presented before the court, these guidelines are used to prevent ‘junk’ science from being exploited in the courtroom (Smith et Bace, 2002):

- Have the procedures (forensic procedures and techniques) been published, preferably in a journal?
- Has the professional community accepted these published procedures?
- Have the procedures been tested?
- What is the error tolerance rate of these procedures?

Reliability and authentication are much related, but both are distinct concepts. The purpose of reliability is to establish whether evidence is what it purports to be, while the authentication is to ensure that the admitted evidence has not been tampered with. For example, if there is video footage of a murder, even if the footage is authentic, meaning it was not tampered with, the prosecutor must prove the video was reliable (i.e., that this video footage actually depicted this particular murder).

Such factors aid the role players to legally complete their CoC documents and improve their contents (i.e., all information provided in this document and describing each forensic phase should respect these common elements). Another level should be fulfilled including: the thoroughness in verifying the origin of this information, where this information came from, how it was collected (reliability), who collected this information, what this information is (authenticity), when it was collected, and why it was collected (relevance). Thus, five “Ws” and one “H” questions should accompany all recorded information in order to build trustworthiness among role players and the judge. The ability to track the origin of information is a key component in fostering

and demonstrating trustworthiness, which is required finally for the admissibility of any digital evidence.

Secondly, from the side of the web itself, the web is a decentralized system full of information provided by a vast number of instruments in every discipline of science and diverse open sources of varying quality. In order to make effective use of the web, provenance metadata should be accompanied by the data itself to describe how data is collected and processed (i.e., 'Who' created and published the data and 'How' the data are published, etc.). This information provides the means for quality assessment for different web resources such as documents, services, ontologies, and datasets. Such resources can also be queried, exploited to reason and consumed to identify their outdated information (Bonatti et al., 2011).

Provenance metadata are not only used to assess data quality but can also support a number of uses (Goble, 2002; Pearson, 2002; Cameron, 2003), such as audit trail (i.e., to determine resource usage and detect errors in data generation), attribution (i.e., establish ownership of data and enable its citation), or/and informational purposes (i.e., using metadata to browse and provide a context to interpret data and more supplementary information related to the data). In addition, because the data on the web is vast, the need for automated processes to annotate all of them is increased (Berry et al., 2003). Interestingly, this was also the main concern of the International Provenance and Annotation Workshop Series (IPAW)¹⁵.

Hence, CoC forensic information should also include provenance metadata. Such metadata can be exploited to give the judge more information about the CoC such as its provenance, its completeness and its timeliness. This information strengthens the provenance dimension of the published data.

¹⁵ <http://www.ipaw.info/>

According to the literature, various methodologies are supported by the semantic web to integrate provenance information to the published data. The straight-forward approach following the Linked Data principles is to use the URI of the RDF document, the data is retrieved from, as subject for statements about its provenance (i.e., meta data should be represented as RDF triples describing the document in which the original data is contained) (Eckert, 2013). Such methodologies can be classified into three main categories. The first category uses the provenance vocabularies of the semantic web (Brickley et Miller, 2014; DCMI, 2015). The second one is to use the Open Provenance Model (OPM) (Moreau et al., 2011). The last category uses the Named Graph (NG) for RDF triples to add provenance metadata about each group of triples. Several provenance vocabularies types are listed in (Hartig et Zhao, 2012).

2.3.1 Provenance vocabularies

Widely deployed provenance vocabularies are the Dublin Core (DC) (DCMI, 2015), Friend of a Friend (FOAF) (Brickley et Miller, 2014), etc. considered as built-in vocabularies on the semantic web, which contain predicates that can provide extra information related to the published data. The objects of these predicates can be represented by URI (e.g., dereferenceable resources) or literal/terminal identifying such objects. Another provenance vocabulary provided in (Hartig, 2009; Hartig et Zhao, 2010) describes how provenance metadata can be created and accessed on the web of data.

All vocabularies presented in the semantic web can express the quality and trustworthiness of any published data.

Trust is a term with many definitions and it is always equated with provenance, but both terms are not the same. The former is derived from provenance information, and

it is subjective and depends on the context, while the latter is used to address the verification of an identity to access an entity. In many cases establishing trust for an entity involves analyzing its origin and authenticity.

Provenance and authentication are often conflated to establish trust. For example, a publisher for a document may also need a digital signature to be authenticated and verified by a third party. (i.e., an example has been provided in Figure 2.4).

2.3.2 Open Provenance Model (OPM)

During a session on provenance standardization in 2006, the provenance research community raised a challenge to understand better the capabilities of different systems, the representations they used for provenance, their similarities, their difference. At this time the first provenance challenge was born to provide a forum for the community to understand provenance systems. After that this was followed by second provenance challenge aiming at establishing inter-operability of systems, by exchanging provenance information. In June 2008, the first OPM workshop was held to discuss some requirements to allow provenance information to be exchanged between systems, allow developers to build and share tools that operate on such provenance model.

The Open Provenance Model (OPM) is a more expressive vocabulary that describes provenance in terms of entities such as agents, artifacts, and processes (Freire et al., 2008; Moreau et al., 2011). Simply, its main objective is to capture dependencies between these entities by constructing provenance graph. Therefore, nodes, artifacts, processes or agents, can be connected by directed edges. An edge represents a relation between its source, denoting the effect, and its destination, denoting the cause. An artifact is generated by a process; a process used an artifact; a process is triggered by another process.

Another work provided in (Zhao, 2010) explains the Open Provenance Model Vocabulary (OPMV), which implements the OPM model using lightweight OWL and assert the OPM concepts. Open Provenance Model Vocabulary OPMV can also be used with other provenance vocabularies, such as DC (DCMI, 2015) and FOAF.

2.3.3 Named Graph (NG)

Whilst many authors advocate the use of semantic web technologies (i.e., vocabularies, Light weight ontologies), the work in (Carroll et al., 2005) proposed Named Graphs (NG) as an entity denoting a collection of triples. The idea of the named graph is to take a set of RDF triples and consider them as one graph, assigning to it a URI reference.

Thus, URIs are used to identify collections of statements. Triple store or RDF store is a database for the storage and retrieval of triples through queries of the semantic web. Adding a name to the triple makes a “quad store” or named graph.

In addition, URI can be assigned to a set of triples, and treat this set as a subset based on its graph identifier (i.e., graph URI). Extending the RDF model from triple to a quad is useful when managing RDF dataset, such as tracking provenance of RDF data (i.e., track the metadata associated to this URI), versioning (i.e., add more description to the URI, label and description), or may also add extra semantic to the URI identifier using different vocabularies of the semantic web.

For example, if we have an RDF graph containing two set of triples (see Figure 2.8):

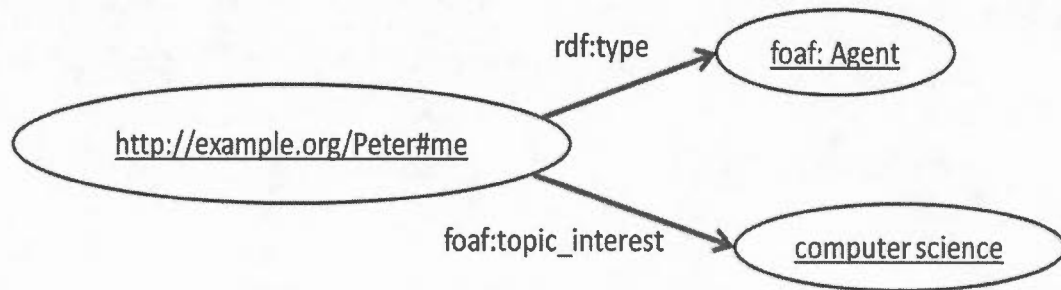


Figure 2.8 RDF triples

The first triple consists of URI subject of type “*foaf:Agent*” and the second triple is the same URI subject that has a computer science value as an object with a predicate “*foaf:topic_interest*”. Both triples can be named together using as a single URI identifier as shown in the next figure:

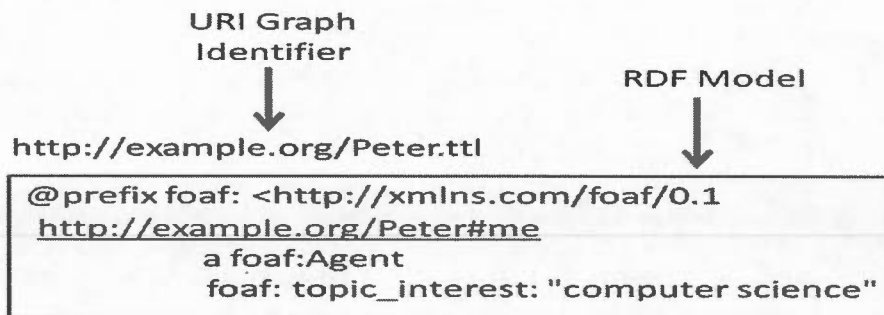


Figure 2.9 Named Graph for RDF model

As shown in Figure 2.9, the RDF model contains two triples. Both triples are described using the vocabulary of FOAF. Metadata can not only be added inside the graph. It can also be used to describe the URI identifier of the graph itself, as shown in the next figure:

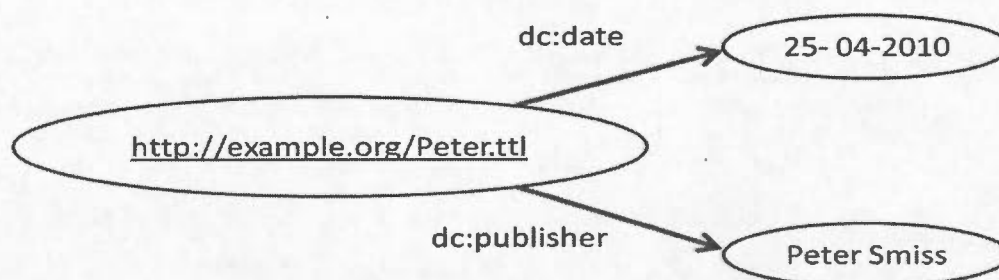


Figure 2.10 Graph identifier with metadata

Adding metadata to the URI graph identifier facilitates the graph management or facilitates the management of a set of triples assembled within this graph. For example, in Figure 2.10, the property terms *dc:date* and *dc:publisher* are used from the DC vocabulary (DCMI, 2015). One provides the date and the other provides the publisher name. Therefore, these metadata will be applied automatically to each triple inside the graph. Thus, the first and the second triple will also have the same date and publisher name.

Therefore, the NG is useful to the consumer to navigate and access provenance metadata related to certain sets of triples and to get more description about them. Another example for that is the LDspider (Isele et al., 2010), which allows crawled data to be stored in quad store using the named graphs data model.

In addition, as the SPARQL is widely used for querying RDF data, it can also be used in the named graph to query single or sets of named graphs. Recent work published in (Omitola et al., 2011) allows publishers to add and trace provenance metadata to the elements of their datasets. This is presented through the extension of the VoID (Vocabulary of Interlinked Datasets) vocabulary into “*voidp*” vocabulary (i.e., lightweight provenance extension for the void vocabulary) (Alexander et Hausenblas, 2009). VoID is an RDF Schema vocabulary for expressing metadata about RDF datasets. VoID is used to relate publishers and users of RDF data and to express

general metadata based on DC, access metadata (describe how RDF data can be used using various protocols), structural metadata (describe the schema of datasets and is useful for tasks such as querying and data integration) and links between datasets (describe how multiple datasets are related and can be used together). Also, the VoID vocabulary considered different properties such as dataset signature, signature method, certification and authority, in order to prove the origin of a dataset and its authentication.

To summarize, the state of the art in this section discussed two important points. The first point concerns how legal documents prepared by the role player can be admissible in the court of law. The second point discussed different provenance technologies that can be used to add extra information to the published data which can be useful in the context of CF.

2.4 Judges awareness of the digital evidence

Judges awareness of digital evidence is related to the fact that judges do not usually have the technical knowledge related to the field of ICT (Kessler, 2010). Judges should have different means to consume and understand the published data related to the digital evidence. The state of the art related to this problem concerns different consumption patterns that can be used by the consumer of the LD to navigate between different resources and to expand (i.e., dereference, “follow-your-nose” style) resources to discover, get and understand the represented information.

From the state of the art of consumption application on the semantic web, there exist four different patterns to consume any published data. As mentioned, LD is a style of publishing data that makes it easy to interlink, discover and consume them on the semantic web. The main way to publish LD on the web is to make URIs that identify

data items dereferenceable into RDF descriptions. Consumers can use four different patterns to consume the information through browsing, crawling, querying and reasoning.

Also, in this section, Linked Education (LE) will be discussed (Dunkel et al., 2006; Evangelia et al., 2011). Nowadays, *e-learning* researchers are trying to exploit the LDP to establish a well-interlinked data for the education domain. This era of research will not be used in the framework, but it should be mentioned, because it can be considered to extend the framework with more educational resources in a future work.

2.4.1 Browsing pattern

Browsing is like the traditional web browsers that allow users to navigate between HTML pages (see Figure 2.2, web of documents). The same idea applied for LD to interact with the web of data (Heath, 2008), but the browsing is performed through the navigation over different resources by following RDF links and downloading them from a separate URL (Quan et Karger 2004) (e.g., RDF browsers such as Disco¹⁶, Tabulator¹⁷, or OpenLink Browser¹⁸) (see Figure 2.3, web of data).

¹⁶ <http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/disco/>

¹⁷ <http://www.w3.org/2005/ajar/tab>

¹⁸ <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/>

2.4.2 Crawling pattern

RDF Crawlers are developed to crawl LD from the web by following RDF links. Crawling linked LD is a search using a keyword related to the item in which consumers are interested (Cheng et Qu, 2009) (e.g., SWSE (Hogan et al., 2013) and Swoogle (Ding et al., 2004)).

2.4.3 Querying pattern

Consumers can also perform extra search filtering using query agents. This type of searching is performed when SPARQL endpoints are installed, allowing expressive queries to be asked against the dataset (Hartig et al., 2009). Furthermore, a VoID vocabulary (vocabulary of interlinked datasets) (Alexander et Hausenblas, 2009) contains a set of instructions that help VoID users. By following them one can succeed in his/her discovery and exploitation and usage of LD datasets through dereferenceable HTTP-URIs (navigation) or SPARQL endpoints (searching), using SPARQL protocol ("*void:sparqlendpoint*") or URI protocol.

As mentioned in Section 1.2.3, judges are more specialized in the legal domain and know more about law procedures. Many do not have ICT skills. The solution that has been proposed in the literature is to organize a training program to educate judges about the field of ICT (Kessler, 2010). This dissertation argues against this solution's direction and will provide *e-CoCs* that can be consumed using different consumption patterns.

2.4.4 Reasoning pattern

The filtering of information is not restricted to extracting explicit information that is stored physically in RDF datasets. It can go beyond. Reasoning pattern can infer implicit information from the RDF triples. As the number of lightweight ontology constructors describing proprietary terms increases, the possibility of inferring extra information also increases (see Section 2.2.1.4).

2.4.5 Linked Education (LE)

Since 2001, when Tim Berners-Lee presented the semantic web as a web interpretable by machines, the researchers of e-learning have been trying to exploit semantic web technologies for *e-learning*. They provided several works on this research. For example, some works (Dunkel et al., 2006; Evangelia et al., 2011) underlined the advantages of using the semantic web for representing the learning object metadata. Other works provided the use of ontologies to describe the contents of learning resources or modeling an e-learning environment by means of a multi agents system (Dietze et al., 2012; Dietze et al., 2013; Keßler et al., 2013).

However, despite these proposals, different learning repositories are still isolated from each others. The Technology Enhanced Learning (TEL) is a new synonym for *e-learning* and refers to technology enhanced classrooms and learning with technology in order to enhance the *e-learning*. It has focused on the interoperability (integration) and reuse of different learning resources and data on the web. They tried to alleviate the great challenge about the heterogeneity of such resources and data. Their works were concentrated on two dimensions: the metadata scheme (e.g., LOM and ADL SCORM (Sharable Content Object Reference Model (SCORM), 2004) and the interface mechanism (e.g., OAI-PMH AND SQI (Van de Sompel et al., 2001;

Lagoze et Van de Sompel, 2003)), which are used to support the interoperability and share different resources on the web in an open way. Despite these works, the integration process is not totally accomplished, and it will be costly to integrate all repositories together.

Nowadays, the researchers of the TEL field find that the LDP is a fertile land to represent and integrate different educational resources. Their works is concentrated on adopting the interface mechanism and metadata scheme into the LDP (Schatz, 2007; Farouk et Ishizuka, 2012).

As been discussed, the LDP provide the “How” to create a semantic web. LDP provide well-established principles and vocabularies based on the technology stack (e.g., URLs, HTTP, and RDF) that facilitate the data interoperability, accessibility and reusability. These features can ultimately be leveraged to construct rich and well-interlinked data for the educational domain. A new research field has emerged called the ‘Linked Education’.

2.5 Security of COC information

The CoC documents must be affixed securely when they are transported from one place to another. Usually, this occurs by sealing the envelope containing the tangible documents. This will not be the same for the information that will be represented to be consumed by computers. Some security algorithms should arise to accommodate the digital nature of information.

In the literature, there exist vast security algorithms. The most related work in literature to the LD was provided in (Rajabi et al., 2012). In their work the authors explained how Public-key Infrastructure (PKI) is used to achieve the trustworthiness of LD and how different datasets are exchanged in a trusted way. As well, the work

provided in (Cobden et al., 2011), outlined in a vision paper the need to have an access restriction on the LOD. Each work apart does not provide the complete picture to realize the LCD using PKI.

In (Rajabi et al., 2012), the work explains how the PKI can be used to secure the resources of LD but did not put the scope on how such stuff can be implemented and applied. This work can bring out a new epoch of research related to the counter part of LOD, Linked Closed Data - LCD, where the publisher would take steps of imposing access restrictions to protect his information.

However, in (Cobden et al., 2011), the work outlined the need of the LCD in certain fields (e.g., business and finance), but did not refer to the PKI solution or how the LCD can be realized. This dissertation complements and completes the half picture of both works, by explaining how the PKI and digital certificates are used to restrict the access of resources in the LD cloud while keeping the resolvability of such resources, to create LCD.

Both works did not provide a solution to secure resources while maintaining their resolvability. This statement is still an open debate. In several situations, URI/URL resources need to obey some access restrictions, where a specific set of people are those who are authorized to access such resources. LDP should be bended to realize the adaptation of publishing and consuming the resources on a closed scale without losing the resolvability feature of these resources. Thus, a trade-off question arises in this case: how can we realize the access restriction over certain URI/URL resources while keeping the resolvability feature of the same resources from anonymous consumption? A very good example to elaborate on this idea is the topic presented in this dissertation, where the represented CoC resources should obey access restrictions in order to be shared on a closed scale among role players and the judge, while keeping the resolvability feature of these resources. Chapter 3 will discuss the

possibility to resolve this compromising question and how PKI can be applied to secure the forensic information.

This section also underlines some concepts from literature related to the PKI, especially the digital certificates: what are digital certificates? Their purposes? Their protocols? Their types? How can they be exchanged?

2.5.1 PKI and digital certificates

PKI is a combination of softwares and procedures providing a means to create, manage, use, distribute, store, and revoke digital certificates (Blaze et al., 1999; Kuhn et al., 2001; Barker et al., 2009; Davies 2011). PKI is called Public-key because it works with a key pair: the public-key and the private-key.

A digital certificate is a piece of information that indicates a recognized proof of a person's identity (e.g., a passport). It uses the key pair managed by the PKI to exchange securely the information in order to create trustworthiness among data provider and data consumer in a network environment (Entrust, 2010) (i.e., trustworthiness occurs when the receiver is reassured of the identity of the sender. As mentioned, it is known as non-repudiation).

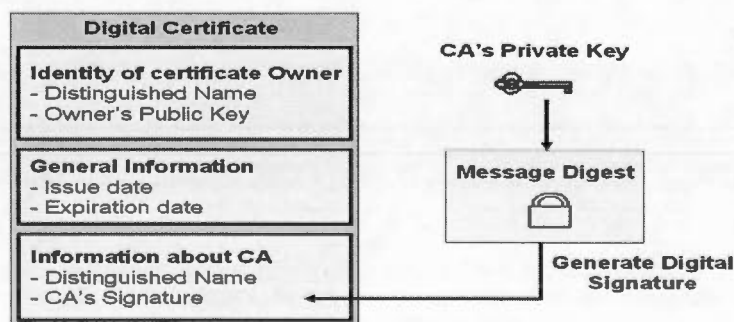


Figure 2.11 Digital certificate (Davies 2011)

Any certificate (see Figure 2.11) contains the identity of the certificate owner, such as the distinguished name, and information about the Certification Authority (CA - issuer of certification), such as CA's signature of that certificate as well as the expiration date and the certificate's issuance date (Perlman, 1999).

A digital certificate alone can never be proof of anyone's identity. A third trusted party is needed to confirm and sign the validity of each certificate and share securely the cryptographic key pair. This party is called "Certification Authority" (CA).

Since a CA (e.g., VeriSign Inc., Entrust Inc., Enterprise Java Bean Certificate Authority-EJBCA, etc.) relies on public trust, it will not put its reputation on the line by signing a certificate unless it is sure of its validity, which makes them acceptable in the business environment.

All digital certificates provide the same level of security, whether they are created by a well-known issuer, or by an unknown one. Usually, the information providers request their certificates from well-known parties when they provide services and information to large segments in society.

2.5.2 Purposes and advantages

A digital certificate has various security purposes and advantages that can be used to (Kuhn et al., 2001):

- Allow only the authorized participant (sender/receiver) to decrypt the encrypted transmitted information (i.e., encryption).
- Verify the identity of either sender or recipient (i.e., Authentication).
- Keep the privacy of transmitted information only to the intended audience (i.e., privacy/confidentiality).

- Sign different information using a signature algorithm (e.g., Ron Rivest, Adi Shamir, and Leonard Adleman (RSA) (Rivest et al., 1983), Digital Signature Algorithm (DSA) (Alajbegović et al., 2006), etc.) in order to ensure the integrity of information and confirm the identity of the signer of such information (i.e., digital signatures). Digital signatures also solve the non-repudiation problem by not allowing the sender to dispute that he was the originator of the sent message.

2.5.3 Protocols

In the field of ICT, the digital certificate is called “SSL/TLS certificate” because it uses two essential protocols: the SSL and the TLS¹⁹. The former is the short version of the Secure Socket Module. This protocol is used to describe a security protocol underlying a secure communication between a server and a client.

After upgrading this protocol with some encryption standards, the protocol got another acronym called TLS, which is standing for Transport Layer Security. Both protocols are based on the public-key cryptography (Perlman, 1999). They are used to establish a secure connection over the HTTP. Classically, the HTTP establishes an unencrypted connection without using the SSL and TLS (i.e., if there is some intruder around monitoring the communication between server and client, he can come with all plain data packages of such transferred data). HTTP is then extended to HTTPS to secure the connection and encrypt all the transferred data with the SSL (i.e., HTTP + SSL/TLS = HTTPS) (Request For Comment (RFC), 1999).

¹⁹ <https://www.evsslcertificate.com/ssl/description-ssl.html>

2.5.4 Types and exchanges

There exist three types of digital certificates. Figure 2.12 presents an abstract scenario where Alice and Bob want to share information over a secure connection (i.e., HTTPS).

Firstly, Alice and Bob should determine a third trusted party called the CA. The latter is responsible to issue SSL/TLS certificates for both of them so that they can identify themselves to one other. CA issues two types of certificates: server certificates and client certificates.

- **Server certificate:** this certificate is issued by the CA and is used by Alice (i.e., suppose that she is the owner of the information) to identify herself to her authorized clients, like Bob. When Bob tries to access this server, he will be sure that he accessed the right one. If Alice fails to identify herself, Bob will not trust Alice's information.

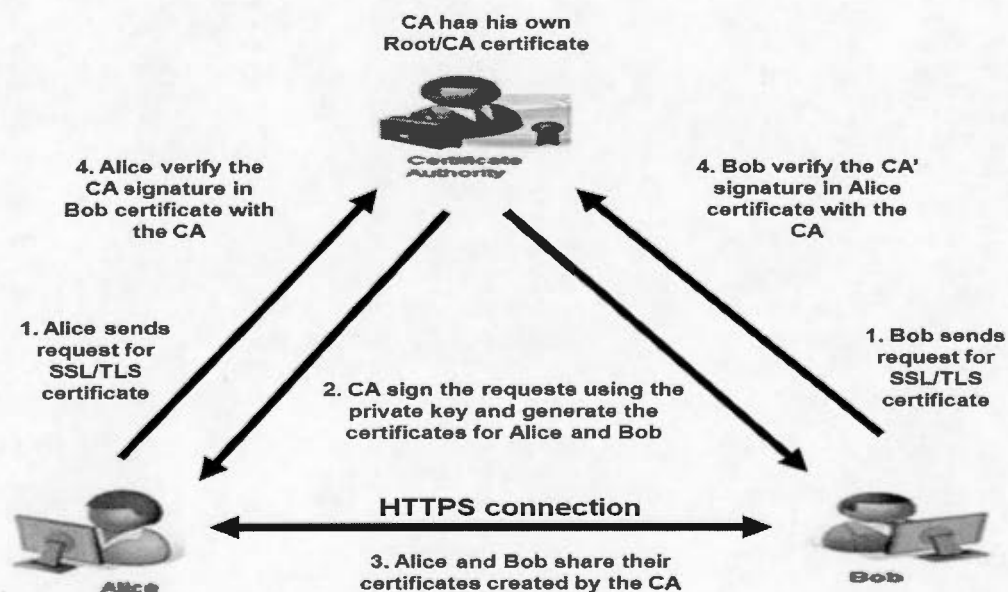


Figure 2.12 Sharing SSL/TLS certificates (Gayed, Lounis et al. 2014a)

- Client certificate: the CA issues this certificate, and it is used by Bob (i.e., suppose he is the consumer of Alice's information) to identify himself to Alice. Alice will not allow anyone to access her information unless he has a certificate known by her.
- CA certificate: the third certificate is the certificate of the CA itself. Sometimes it is called a "self-signed" or "root certificate", because it is the CA itself who will sign its certificate. The CA uses this certificate to sign the certificate requests received from the clients and servers. In addition, this type of certificate answers the question of how Alice and Bob confirm the identities of each other. Alice would know that Bob is the right person by verifying that his certificate is signed by the common trusted part authority (CA), and vice versa. Both identify themselves through the CA certificates.

From the definitions mentioned above, we notice that there is no distinguishable difference between the server certificate and the client certificate; both use the certificates to identify themselves to the other. The only difference that distinguishes both concerns who is providing the information and who will consume it.

2.6 Conclusion

This chapter explained different concepts and technologies related to the research problems. The first part started by introducing the semantic web and how it uses the LDP to create web of data. The latter uses the RDF as a standard format to represent link and interoperate information on the semantic web. It depicted how RDFS constructors and some primitives from OWL can be used to create a lightweight ontology, how to define new/proprietary terms and how these constructors can be used to infer implicit information from the store of RDF triples.

This chapter discussed how the LDP is used to create the semantic web and how it can be used to represent information using the light weight ontology and define proprietary terms. This will be also useful to represent forensic information. This idea will be elaborated on Chapter 3 through mentioning several advantages of using LDP to represent such information and how lightweight ontology corresponds to forensic phases and tasks.

The second part was about fostering trustworthiness among role players and judge. The state of the art mentioned in this part is related to provenance technologies of the semantic and how to foster information using different provenance vocabularies and techniques. Chapter 3 will discuss how the provenance information can be adapted and used to add another dimension to the forensic information. This dimension will be considered as a supplementary metadata that will aid judges to know the origin of the published information.

The third part depicted different consumption patterns that can be used to consume any published data on the semantic web. As mentioned, judges usually do not have ICT skills and as a result they, may not be able to understand or take the proper decisions toward the presented digital evidence. Hence, the main objective of this dissertation is to let the judges understand digital evidence presented to them. Whatever technology is used to accomplish this objective, it should also be clear in its mechanism and methodologies. For example, judges cannot consume published data through SPARQL query language since this consumption pattern necessitates the awareness of semantic web and technical skills to write SPARQL code.

Finally, the last part in this chapter discussed the PKI and different advantages of digital certificates. The next chapter will discuss how digital certificates can be adapted to LOD in order to consume the forensic information on a closed scale among role players and the judge (LCD). Requests of certificates will be the responsibility of a neutral side that is responsible to select the proper issuer institution

to issue the server certificate for its CF-CoC system and the client certificates for the judge and role players.

CHAPTER III

RESEARCH METHODOLOGY

3.1 Introduction

This chapter discusses the research methodology to build a system that can address all research problems. The first problem is that the system must support the possibility to transform the CoC from the tangible document into electronic information that is consumable by people and machines. Chapter 2 discussed how the information is expressed on the semantic web. The question is: how can such principles be exploited to represent forensic information, and what are the advantages of using such representation in cyber forensics?

The second problem concerns trustworthiness that must be built among judges and role players. This objective won't be reached unless the judge knows from where such represented information came from, and when, who, where and how the represented resources are published. Is the semantic web able to provide supplementary information to annotate the published resources (Berry et al., 2003) that makes the *e*-CoC admissible in a court of law?

The third problem is related to how the judge can manipulate this electronic representation instead of the tangible documents. Does the judge own the necessary information on how to consume this electronic representation? At the same time, can this representation add extra information to help the judge understand the digital evidence?

The last problem deals with the possibility to use the web aspects, which are used to publish public information on the web; to publish and represent information that should be used on a closed scale among role players and judges. Is it possible to bend such principles to be used on a closed scale and secure the published resources?

The approach that we present must integrate different solutions to these problems and lead to a system able to address all challenges mentioned in Chapter 1.

3.2 Representing CoC using LDP

The first problem presented in this dissertation is the need to transform the tangible CoC into a form that accommodates the digital technologies, especially so that such documents contain information about digital evidence. The first hypothesis proposed to accomplish this radical transformation is to use the semantic web to represent and manage the tangible CoC.

This hypothesis states that the semantic web can be a fertile land to create interlinked *e-CoCs*, which are readable and consumable by people and machines, and the forensic information resulting from a forensic tool can be interoperable with these interlinked CoCs (Gayed et al., 2012a, 2012b). Before going further, the next part will highlight the main reasons to exploit knowledge representation itself (i.e., regardless of the how) as a means to transform this information into an electronic format.

Knowledge representation has been persistent at the centre of the field of Artificial Intelligence (AI) since its founding conference in the '50s (McCalla et Cercone, 1983; Ringland et Duce, 1988; Shrobe et Szolovits, 1993). This concept is described by Davis et al. through several distinct roles (Davis et al., 1993) a representation plays:

- *A surrogate, a substitute for the thing itself:* each surrogate corresponds to its referent in the real world. Thus, a knowledge representation of a CoC serves as a surrogate to that CoC (e.g., the surrogate of the tangible CoC, which exists in the physical world, is the representation of the *e*-CoC).

- *A set of Ontological commitments:* according to Gruber, “ontology is an explicit specification of a conceptualization”. Conceptualization means that an aspect of the world is described by an abstract model. This model (including concepts, properties and relationships) is described using some formal language, making it consumable by humans and machines. In this context, the representation of the CoC using the LDP will contain simplifications and assumptions according to the perspective (conceptualization) of the role player to model different forensic entities and their relationships using well-defined vocabularies (unambiguous) from the semantic web. This point will be explained in Section 3.2.1, point 7.

- *A fragmentary theory of intelligent reasoning:* AI uses knowledge representation to enable some automated reasoning. Different logic formalisms are used for knowledge representation to support reasoning and inferences. Recently, different works have been provided for linked data reasoning (Bonatti et al., 2011; Corby et al., 2012; Farouk et Ishizuka, 2012; Freitas et al., 2012). As mentioned in Chapter 2, this will be very useful for judges to use the machines to infer implicit information from the represented information.

- *A medium of human expression:* The role player will use knowledge representation as a medium to express different concepts about the tangible CoCs (external world) for the machine or for other people (i.e., judges). The

knowledge representation allows role players to provide more details about the CoC. The semantic web is rich with different vocabularies and provenance metadata that allow the role player to express CoC information.

The state of the art in Chapter 2 showed that the semantic web uses LDP as a means to create web of data. Therefore, next subsection 3.2.1, discusses advantages of using the semantic web and its principles to represent forensic information. Also, it will explain why the forensic format resulted from a specific forensic tool can also be interoperable with this represented information.

After discussing these advantages, the subsection 3.2.2 discusses how forensic phases and tasks are corresponding to the ontology structure. Section 3.2.3 explains how this forensic information is described on each level: for instance, how to create a forensic phase and then how its tasks are described using proprietary terms, as well as how these terms are selected and defined to describe different resources associated to each task. Finally, how created terms are used to publish forensic information is also described.

3.2.1 Why LDP for representing forensic information?

There are several advantages for representing the forensic information using LDP (Gayed et al., 2013b):

1. CoC and LDP have common features. The most common feature is the interlinked nature. This feature is indeed shared by CoC and the RDF data model. The nature of CoC is characterized by interrelation/dependency of information between different phases of the forensics process. Each phase can lead to another phase. This interrelation fact is the basic idea (“follow-your-nose” style) over which the LD is published, discoverable and significantly

navigated using RDF links. RDF links in LDP would not only be used to relate the different forensic phases together. It can also be used to assert connection between the entities described in each forensic phase. Also, RDF typed links enable the data publisher to state explicitly the nature of connection between different entities in different phases or same phase, which is not the case with un-typed hyperlinks used in HTML.

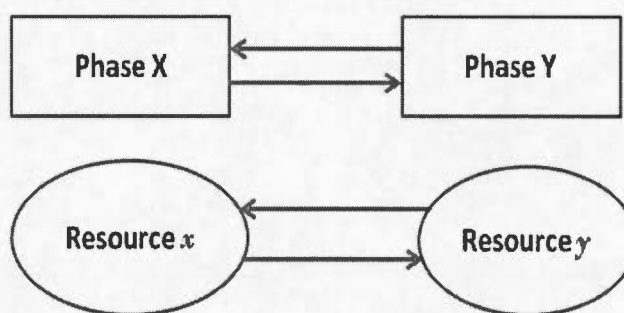


Figure 3.1 Correspondence of forensic phases and LD resources

As shown in Figure 3.1, as the forensic phases are interlinked together, the resources of LD can also be interlinked using forward and backward links.

2. LDP enable links to be set between items/entities/resources in different data sources using a common data model (i.e., RDF) and web standards (i.e., HTTP, URI, and URL). As well, if the CoC is represented using the LDP, the items/entities in different phases of a forensic process can also be linked together. This will generate a space in which different generic applications can be implemented:
 - a. Browsing applications: will enable judges and role players to view data from one phase and then follow RDF links within the data to other phases in the forensics process.
 - b. Search engines: judges and role players can crawl the different phases of the forensics process using different search keys (keywords).

- c. Query published information: representing information using LDP allows the judges and role players to also trigger sophisticated queries to infer implicit information. However, they will not write query code by themselves, which will be discussed later in consumption patterns.

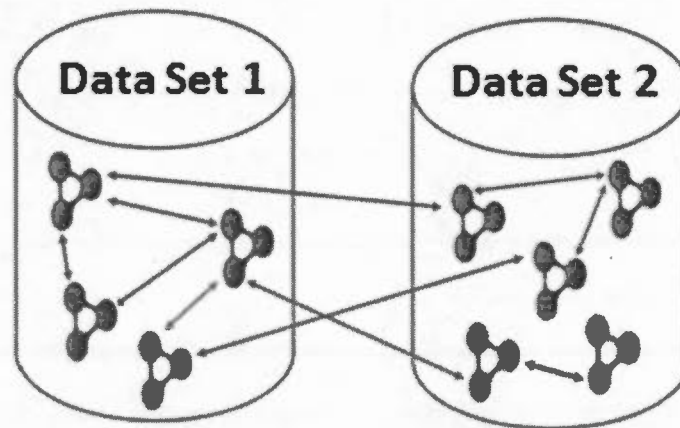


Figure 3.2 Interrelation between two datasets

3. Ontology can contain at least one dataset. As shown in Figure 3.2, a dataset contains different RDF triples. Each resource is connected to another resource within the same dataset, another dataset, or to ontology. Interrelation between entities facilitates their consumption using at least one pattern mentioned above. This is also the case between entities/resources of different forensic phases.
4. LD applications that are planned to be used by judges and role players are able to translate any data even if it is represented with unknown vocabulary. This can be realized using two methodologies:
 - a. Firstly, by making the URIs that identify vocabulary terms dereferenceable (i.e., it means that HTTP clients can look up the URI using the HTTP protocol and retrieve a description of the resource that

is identified by the URI), so that the client applications can look up the terms, which are defined using RDFS and OWL.

- b. Secondly, by publishing mappings between terms from different vocabularies in the form of RDF links. Therefore, for any new term definition, the consumption applications are able to provide and retrieve more information describing the provided data.

Both facts allow the judge and the role players to explore and navigate between resources in order to get supplementary information about such resources.

5. Nowadays, RDFS (W3C, 2014) and OWL (Dean et al., 2004; McGuinness, 2004; Van Harmelen et McGuinness, 2004) are partially adopted in the web of data. Both are used to provide vocabularies for describing conceptual models in terms of classes and their properties (definition of proprietary terms). RDFS vocabularies consist of class “*rdfs:Class*” and property “*rdf:Property*” definitions, which allow the subsumption relationships between terms. This option is useful for judges to infer more information from the data in hand using the entailment rules mentioned in Chapter 2. For example, as mentioned, RDFS uses a set of relational primitives (e.g., “*rdfs:subClassOf*”, “*rdfs:subPropertyOf*”, “*rdfs:domain*”, and “*rdfs:range*”) that can be used to define rules allowing additional information to be inferred from RDF graphs.

Also, OWL extends the expressivity of RDFS with additional modeling primitives that provide mapping between property terms and class terms at the level of equivalency or inversion (e.g., “*owl:equivalentProperty*”, “*owl:equivalentClass*”, “*owl:inverseOf*”). This will be useful for the role player to map between terms. This occurs when a role player finds that some of the terms

that he has created are similar or equivalent to other terms created by another role player in another forensic phase.

OWL and related vocabularies including provenance are not yet fully adopted on web of data, but soon the full adaptation will be achieved (Zhao et al., 2010; Glimm et al., 2012). This will be a great advantage to add more property and class terms to the ontological dimension of the LD, and therefore, provide useful and descriptive information.

6. By using LDP to represent the CoC, the latter will be enriched with different vocabularies such as, Dublin Core (DC) (DCMI, 2015), Friend of a Friend (FOAF) (Brickley et Miller, 2014) and Semantic Web Publishing (SWP). In addition, vocabulary links are one type of RDF links that can be used to point from data to the definitions of the vocabulary terms, which are used to represent the data, as well as from these definitions to related terms into other vocabularies. This means that there is a mix of data to the definitions. This mixture is called “schema” in the LD and contains distinct terms imported from different vocabularies to publish the data in question. This mixture may include terms from widely used vocabularies, in addition to proprietary terms. Thus, we can have several vocabulary terms to represent the forensics data and make it self-descriptive (i.e., using the two methodologies mentioned in point 3) and enable LD applications to integrate the data across vocabularies and enrich the data being published.
7. The forensic information should not confuse judges in courts. Contradictions and heterogeneity need to be avoided in the information provided to the judges by the role players. LD tries to avoid heterogeneity by advocating the reuse of terms from widely deployed vocabularies (same agreement of ontology – as being mentioned ontology is an explicit specification of a

conceptualization) in order to increase homogeneity of descriptions and consequently easing the understanding of these description (Jentzsch, 2013). As mentioned, the widely deployed vocabularies of the semantic web do not cover all domains. However, linked data sources still cover a wide range of topics, but they do not cover all aspects of these topics (nowadays, there exist at least 369 different vocabularies on the web of data²⁰). An example is the forensic field, where role players will commonly define their own new custom terms (proprietary terms) and mix those terms with the widely used ones to explain and describe more specific aspects and publish all the content from his forensic investigation. Therefore, it is a great advantage to use LDP to represent such information.

8. As mentioned in point 1, a forensics process contains several phases that depend on and relate to each other. Each entity is identified by a URI namespace to which it belongs. An entity appearing in a phase may be the same entity in another phase. The result is multiple URIs identifying the same entity (i.e., same idea as point 5). These URIs are called URI aliases. In this case, LD relies on setting RDF links between URI aliases using the “*owl:sameAs*” that connect these URIs to refer to the same entity. The advantages of this option in CoC representation are:
 - a. Social function: investigation process is a common task between different players. The descriptions of the same resource provided by different players allow different views and opinions to be expressed.
 - b. Traceability: using different URIs for the same entity allows judges that use the CoC published data to know what a particular player in the investigation process has to say about a specific entity of the case in hand.

²⁰ <http://lov.okfn.org/dataset/lov/>

The same occurs, not only at the level of URI, but also at the level of terms (i.e., point 4-b). Players of the forensics process may discover at a later point that the build-in vocabularies contain terms similar to those that they have created. Players could relate both terms, stating that both terms actually refer to the same concept using the OWL (“*owl:equivalentClass*”, “*owl:equivalentProperty*”) and RDFS vocabularies (“*rdfs: subclassOf*”, “*rdfs:subPropertyOf*”).

9. Semantic web contains also provenance metadata (e.g., DC and FOAF) that can be published and consumed in the web of data (Hartig et Zhao, 2010). These metadata can answer the 5Ws and 1H questions at the level of the data origin (see Chapter 2). These vocabularies can be used concurrently with the forensics data to describe their provenance and answer the questions of the forensic investigation.
10. As mentioned in Chapter 2, the work presented in (Giova, 2011) presents the idea to translate the AFF4 into RDF resources in order to improve digital forensics process. The RDF is the standard format of the semantic web, thus the translation of the AFF4 into RDF model means that the AFF4 will contain a set of triples presented in the same structure used by the semantic web (subject, predicate and object). This will facilitate interoperability between the AFF4 format generated by a forensic tool and translated into RDF format and the forensic information described by the technician from the other side (Gayed et al., 2012a).

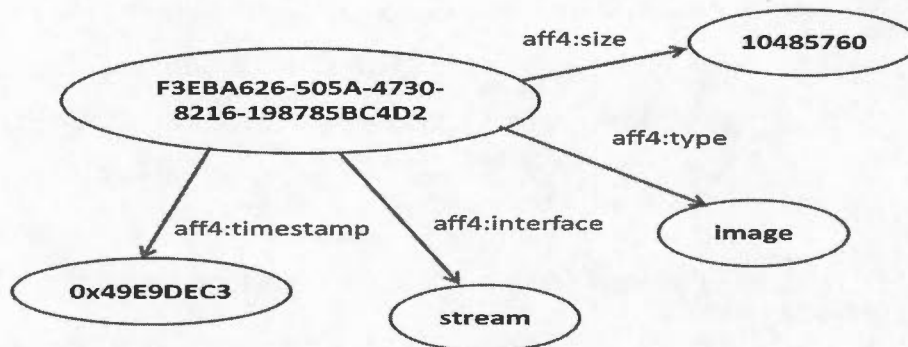


Figure 3.3 RDF model for AFF4 vocabulary

Figure 3.3 depicts the translation of AFF4 format into RDF model²¹. This format contains a name space (vocabulary) called “*aff4:*” where different predicate terms are well-defined such as size, type, interface, timestamp, etc. For example, the main information in this graph is the one related to a suspected file, its size, its type, its interface, etc. Wherefore, this information can be integrated easily with CoC information associated to a forensic phase (Gayed et al., 2012a).

Finally, the LD is the new moderation of the educational research. Constructing a system using LDP has several advantages for the possibility of creating and integrating educational resources. This fact will be useful in a future work for an enhancement of the awareness of judges of the digital evidence, and also for the technicians during information publishing (i.e., this thesis assumes that the technicians own this knowledge). Also these advantages can be considered the reasons that led to the emergence of Linked Education (LE):

- a. *Interoperability*: one of the fundamental of the LDP is the interlinking of data that is based on a set of well established principles and W3C standards (e.g. RDF and SPARQL) and use of URIs, which promote the

²¹ <http://forensicswiki.org/wiki/AFF4>

interoperability between data on the web. This fact allows the construction of a well-interlinked data for the educational domain through the identification of potential links between individual resources.

- b. *Unified Interface Scheme*: as mentioned before, several linked data-consuming patterns with unified interface (e.g., SPARQL) can be implemented to provide added-value services for the consumer (i.e., judges and role players).
- c. *Interaction and collaboration*: as consequences from (b), the LDP promote that the providers of data (i.e., role players) and consumers are able to interact.
- d. *Dereferenciability*: resources represented using the URIs are dereferenceable, where more information describing such resources can be retrieved. This improves the subject matter.

3.2.2 Correspondence between forensic phase and ontology

It is necessary to explain how a forensic phase is corresponding to an ontology. As shown in Figure 3.4, each forensic phase will have a corresponding lightweight ontology. Each lightweight ontology has a set of categories, which will be equivalent to a set of forensic tasks. A category in the vocabulary should be described using a set of terms. These terms are the proprietary terms describing a forensic task.

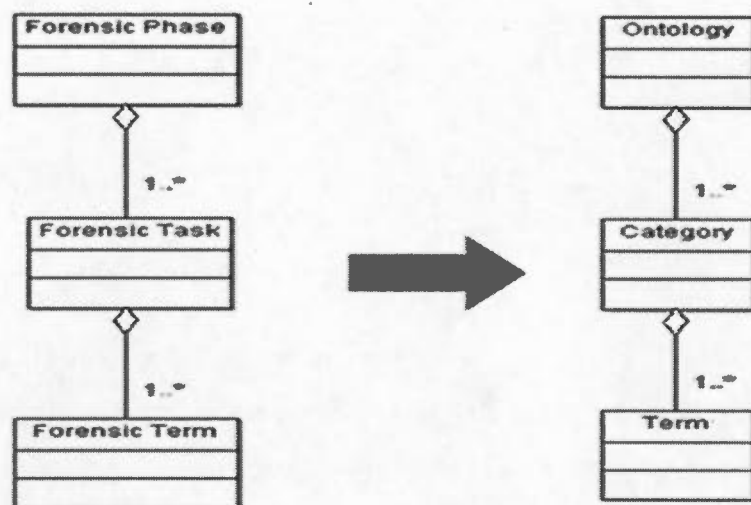


Figure 3.4 Correspondence between cyber forensic phase and Ontology
(Gayed et al., 2014b)

Ontology (vocabulary) may contain a set of categories. Each category contains a set of terms to present this category. For example, the well-known vocabulary FOAF (Brickley et Miller, 2014), contains a set of categories like FOAF Basics, “*Personal Info*”, “*Online Accounts*”, “*Projects and Groups*”, and “*Documents and Images*”. There exist well-defined terms to express each category. For example, the category of Online Accounts is described by a set of terms, some of which are terms of type class (e.g., “*OnlineAccount*”, “*OnlineChatAccount*”, “*OnlineGamingAccount*”, etc.) and others that are terms of type property (e.g., “*plan*”, “*based_near*”, “*topic_interest*”, “*publications*”, “*knows*”, etc.).

Same analogy is used by a forensic phase. For example the “*Acquisition phase*”, it contains a set of forensic tasks needed to accomplish this phase, such as state preservation, backup and copy. These can be described through different terms. For example, for state preservation, some terms can be defined with type property (e.g., “*SN*”) and others will be of type classes (e.g., “*Media*”). For example the Media has

a serial number of type string, where Media is a class and a serial number is property term.

3.2.3 Creating proprietary terms

In a domain like CF, it is rare to find forensic terms or well-known vocabularies describing it, because it is still in its infancy and development. Thus, CF is a domain that requires the definition of new proprietary terms (Gayed et al., 2014b). As shown in Figure 3.4, each forensic task is described using a set of terms. These terms are selected by the technicians and can be of type class or of type property. The root definitions of those terms are defined using well-known vocabularies of the semantic web.

Before creating custom terms, the container and category should be defined first. The container is the lightweight ontology that contains different categories to which these terms belong. The container and all its subcomponents are also custom creations. Custom terms cannot be added to or created for well-defined ontologies (e.g., FOAF, DC, RDFS, etc.). However, they may be appended to another custom ontology created by another technician of the forensic phase. In this way, a collaboration among technicians takes place.

As mentioned in Chapter 2, the selection of terms is a subjective task. Each role player has his own point of view to select and define his terms. Redundancy of terms does not affect the quality of published data due to the two reasons mentioned in Section 2.2.1.5: terms can be dereferenceable and can be mapped.

Terms are not overloaded by different ontology axioms. The RDFS++ constructors will be used to define terms using the vocabularies of the semantic web. Also, terms

will be defined using unique URL and will belong to a unique domain (see the criterion of creating proprietary terms in Section 2.2.1.5).

Also, Chapter 2 depicted the difference between 303 URIs and hash URIs. Hash URIs will be used to identify the forensic resources because they have the advantage of reducing the number of necessary HTTP round trips, which in turn reduces access latency (Sauermann et al., 2011).

The disadvantage of the hash URI approach is that the descriptions of all resources that share the same non-fragment URI part are always returned to the client together, irrespective of whether the client is interested in only one URI or all. In the current context, where judges have limited knowledge about ICT field, it is an advantage to use the Hash URIs versus 303 URLs, because it is better to return to judges all resources that share the same non-fragment URI part.

To create new proprietary term we need to define its properties and relationships, and this is called the terminological definition of the term (i.e., T-Box). After defining the proprietary term, we can use it to publish various triples and this is called the assertion level (A-Box).

3.3 Adding provenance metadata to the *e*-CoC

State of the art in Chapter 2 explained different techniques to add provenance metadata to the published information. Generally, the provenance metadata can be added on the level of T-Box, on the level of A-Box or on both levels. This means that the provenance metadata can be added during the creation of terms or during their usage to publish different information.

The Named Graph technique is the most simple and straightforward approach to add provenance metadata to the published information. It is based on the idea to group a set of triples together and identify them using a URI, allowing descriptions to be made for this set of triples. Hence, technicians can group and manage group of triples together using this approach to describe a forensic task or forensic phase. By grouping these triples, technicians can annotate them using different provenance vocabularies (Berry et al., 2003).

To illustrate how the NG can be applied in this context, we can select any forensic phase from any forensic model. For example, the authentication phase of the Kruse model (Gayed et al., 2013a, 2015) can be represented by a set of triples.

Figure 3.5 indicates an abstract diagram depicting the grouping of triples and naming them by a graph with the integration of provenance metadata (e.g., injection of terms from provenance vocabularies like DC, or FOAF). Each phase will also contain inner and outer links that relate all CoCs to each other.

This figure depicts how a Named Graph (NG) can be applied for a forensic process (e.g., the Kruse model). This figure contains three forensic phases: acquisition, authentication, and analysis. Each of them, expressed as NG, contains a set of triples. These triples can be grouped together and minted using URI reference (e.g., NG for authentication phase is the NG_{Auth} that contains a set of triples grouped and minted by URI). Each NG representing a forensic phase can be associated with one or more metadata terms, which can be imported from well-defined vocabulary (i.e., see also Figure 2.9 and 2.10).

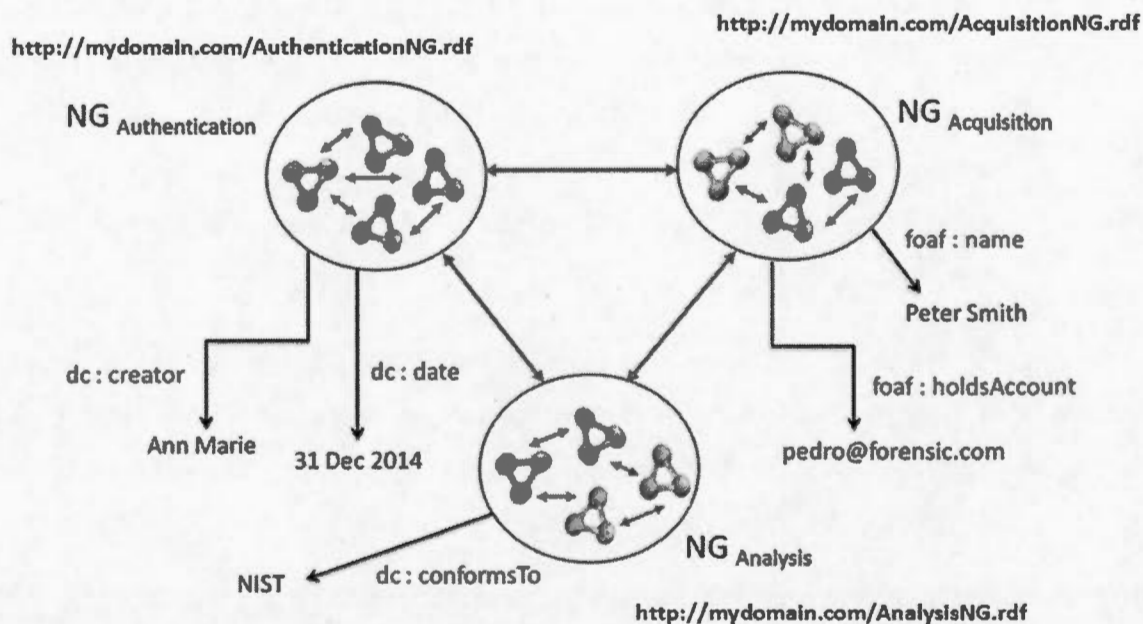


Figure 3.5 Named Graphs for the Kruse Model (Gayed et al., 2013a, 2015)

Table 3.1 summarizes the Figure 3.5 and shows the URIs used to mint each forensic phase.

Table 3.1 Forensic Named Graph

Forensic Phase	Minted to	Provenance Vocabulary
Authentication	http://mydomain/AuthenticationNG.rdf	Dublin Core
Acquisition	http://mydomain/AcquisitionNG.rdf	Friend of a friend
Analysis	http://mydomain/AnalysisNG.rdf	Dublin Core

The provenance metadata determine the origin of published data. Thus, they are considered to be ancillary services that can provide supplementary information to the knowledge domain. This information allows the consumers to find and interpret the origin of different resources. Provenance vocabularies will be integrated within the CF domain to establish trust among role players and judges with respect to the shared information.

In this context, the technicians are responsible to add different provenance metadata to describe their forensic information. Each technician is responsible to provide complete and correct information about the origin and contents of his CoC(s) in order to make such information admissible in a court of law.

Determining the origin of information is crucial on the open share scale (i.e., open science, open government, and intellectual property and copyright), where the consumer needs to ensure who exactly published the represented information. Indeed, the published information needs to be tracked and verified in order to ensure its creditability. Therefore, determining the origin of information being published is mandatory to make the consumer confident towards the information in hand. Usually, this occurs automatically, thanks to software systems. They process and record some basic facts about the terms/resources.

However, on a closed scale, the case is totally different. Sharing forensic information among judges and role players should take place on a closed scale (i.e., LCD, see Section 3.4), whereby a neutral side owns a server where the CF-CoC resides, validates the identities of the role players and judges before participating in the forensic investigation process.

After validation, the neutral side is confirmed by the identities of publishers from the prosecutors and defenders; the publishing and consuming of represented information will be limited among consumer and publishers.

In this context, adding provenance metadata does not necessitate automation and can be done manually by annotating the forensic information. Such metadata will be trusted information since the publishers and consumers are both identified and well-known.

Briefly, the provenance metadata can be added at different levels. They can be added manually during the design of terms (i.e., to describe the term itself), during the publication of terms as a reference in a concrete dataset (e.g., CoC) to use on the semantic web (i.e., to add more information about the data being published) or after grouping a set of triples together and naming them using URI reference as a reference in a concrete dataset (e.g., *e*-CoC).

3.4 Consumption patterns

As mentioned in Chapter 2, there exist four patterns to consume the data. The framework proposed in this dissertation will use these four patterns to aid judges and role players to consume and understand all published resources.

Judges should be separated from the technical details related to these patterns. As mentioned in Chapter 2, browsing on the web of data is the same idea of browsing on the web of documents. Browsing the web of data is navigating through links to discover more resources and this means that the resources are named using URIs and that links are discovered through HTTP. Both are components from the technology stack of LDP. Wherefore, the browsing of RDF resources can be applied to let judges consume these forensics resources.

On the other hand, crawling will allow judges to search different resources using keywords. Crawling will be implemented to let judges search for a specific resource, whether on the level of T-Box or A-Box.

The query pattern is used to retrieve explicit resources from the RDF data. However, reasoning is used to infer implicit information from the RDF dataset. In both cases, it is not necessary that consumers should be aware of how to write a SPARQL code or understand the entailment rules. The proposed framework will implement the reasoning pattern based on the entailment rules discussed in Section 2.2.1.4.

3.5 Adapting Public-Key Infrastructure to Linked Opened Data (LOD)

This section discusses how digital certificates can be applied to LOD in order to publish and consume data on a closed scale (Gayed et al., 2014a). It depicts how digital certificates are used to restrict the access to these resources while keeping their resolvability to discover and navigate among other resources. After doing this task, the published information will be consumed on a closed scale called Linked Closed Data (LCD) (see Section 2.5).

Generally, the digital certificates will be used to serve the neutral side to ensure the identities of the publishers (i.e., role players) and consumers (i.e., judges and role players). The identities of technicians and judges need to be verified before the investigation process and prior to using the CF-CoC system (i.e., before publishers start creating and publishing their forensic information). Also, the identity of prosecutor and defender will be identified once they are engaged after collecting the evidence. Once this is realized, all the published resources will be shared with the authorized parties.

The next paragraphs will depict the adaptation of digital certification to LOD from two sides: (i) the access to resources themselves, and (ii) the access of publishers (i.e., role players) and consumer (i.e., judges).

Referring to Figure 2.5 of the linking open data cloud diagram, several interrelated datasets can be found that use outer and/or inner links. Each dataset is published in a unique domain owned only by the publisher of this dataset over the WWW space. Each dataset contains a set of URI-defined resources that are interrelated within that dataset or to an outer dataset.

Now, imagine that the owner of a dataset wants to publish resources using the technology stack/LDP of the LD (URI, HTTP, and RDF) and have such resources resolvable within the LOD cloud, but at the same time, he wishes to publish them in a manner so that anonymous parties on the web cannot access them.

The idea of both features co-existing, resolvability and access restrictions of resources, resides in the digital certificates. The latter can be used to restrict the resolvability of resources in a one-way manner.

A resource r is forward resolvable in a domain d when this resource explores and discovers other resources on other domains, this means that the resource r is forward deferenceable. A resource r is backward resolvable when other resources on other domains are able to explore and discover the resource r in the domain d , this means that the resource r is backward deferenceable. Wherefore, to restrict access of resources on the web, the access should be forward resolvable not backward resolvable.

The same concept can be applied between datasets/resources in the LOD cloud using digital certificates, where each dataset owns a digital certificate(s). The publisher of the resources can accomplish his publication task through an enhanced technology stack using a secure access protocol (i.e., HTTPS). Therefore, the current technology stack is transformed from (URI, HTTP, and RDF) to (URI, HTTPS, and RDF).

A hypothetical scenario will be as follows: assume on the LD that there is a server (i.e., where publishers publish their information) and there are consumers and both

already have a common trusted party to issue their certificates. The server has a domain name given by an IP (i.e., for simplicity consider this IP corresponds to a domain string name in the LOD cloud²²). The server owner (e.g., the publisher himself) of this domain only wants someone called "*Person X*" to be able to consume the published resources from his domain within the LOD cloud. In this case, the owner restricts access to the resources to only this specific consumer (i.e., X), while keeping the deferenceability of his resources to other resources on other domains. The owner of the server will also be able to move back to his domain using the backward link, because he owns the server certificate for this domain. Any other anonymous party outside this domain will not be able to access the resources of the server. If the server owner wants a new person, else than "*Person X*", to access his resources, this new person should also have a client certificate signed by the same trusted party.

Talking in an LD manner, we can not only consider the server and client side as persons (i.e., server owner, who owns server certificate and "*Person X*" who wants to access these restricted resources), but as datasets or resources within these datasets that can be interlinked together using inner and outer links (i.e., by moving backward/forward from and to the publisher resources). In addition, another important point should be underlined: "*Person X/dataset/resources*" can also react as a server side, if we look at the picture from the inverse direction. This will be explained in the next paragraphs.

²² Domain owned by Tamer Gayed: www.cyberforensics-coc.com

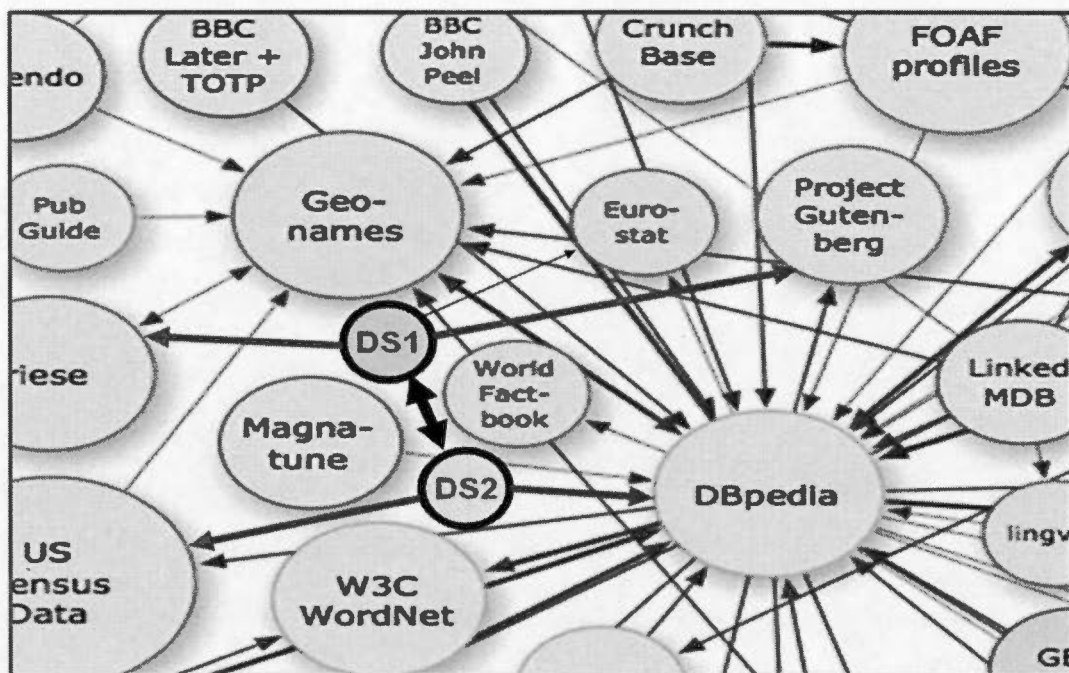


Figure 3.6 Client/Server certificate exchange between two datasets (Gayed et al., 2014a)

Thus, Person X/dataset/resource may also have a server certificate for his/its domain and only allow access to a person/dataset/resource that has a client certificate to his/its domain.

To illustrate on this idea, Figure 2.5 of the LOD cloud is enlarged, resulting in Figure 3.6. Let us consider that there are two datasets DS1 and DS2 residing in two different domains. Each domain represents a dataset. Both of them are interrelated using inner and outer links. As well, both datasets are related with other datasets in the LOD cloud.

To elaborate the idea in terms of datasets, let us consider two datasets: DS1 and DS2. Each of them can be a client and a server at the same time and has client and server certificate. An outer link connects both datasets together in both directions. When DS1 is the client and DS2 is the server, the former will be able to resolve the

resources of the latter. However, any other datasets that do not have client certificates like DS1; will not be able to resolve resources from DS2. This means that the resources of DS2 have access restriction and resolvable only by DS1 or any other dataset that has a client certificate. This is also the same case when DS2 is the client and DS1 is the server. In this scenario, the server and client certificates of DS1 and DS2 made the resources of both datasets forward resolvable, not backward resolvable.

Furthermore, the certificates can not only be used at the level of datasets (i.e., including all resources), but they can also be issued at the level of a specific resource within the datasets. This can be realized by issuing the certificate using one of the three URI patterns provided in Section 2.2.1.

The next part presents a scenario where the neutral side will share digital certificates with judges and role players. The technical details of this part will be explained in Section 6.4 to describe how this scenario can be implemented and realized (Gayed et al., 2013a). Assuming that the neutral side already selected the issuer institution:

1. Role players: technician, prosecutor, and defender, each of them, sends a request to the neutral side that hosts the CF-CoC system, in order to get a client certificate.
2. The neutral side receives the requests and communicates with the CA to issue the client certificates for the role player. Also, the neutral side requests a server certificate for its host and a client certificate for the judge if the latter will be engaged.
3. CA issues the certificates and communicates with role players, neutral side, and judge to provide the requested certificates.

Referring to the same idea presented in Figure 2.12, Alice and Bob are now corresponding to the role players (i.e., clients including judges, prosecutors and

defenders) and just assume that they will have also an intermediate party, neutral side, who is responsible to communicate with the CA. The CA is the certificate authority that the neutral side selects to issue client and server certificates.

3.5.1 Why use the PKI approach?

Aside from the general advantages mentioned in Section 2.5.2, there exist technical reasons motivating us to use the PKI approach rather any secret-key system (Karnin et al., 1983). A lot of secret-key systems proposed in the literature including the Kerberos secret-key (Bellovin et Merritt, 1990), Data Encryption Standard (DES) (Coppersmith, 1994), Advanced Encryption Standard (AES) (Rijmen et Daemen, 2001; Miller et al., 2009), etc.

The PKI approach is designed mainly to secure communication over a non-secure communication channel without having to share a secret key. This is accomplished by generating a public-private-key pair (will be explained in Chapter 6, Section 6.4). Simply, one is used for encryption and the other is used for decryption. Both of them are mathematically related, but no one can infer one key from another. Because a pair of keys is required, this approach is called “asymmetric cryptography”.

In the classic approach (i.e., with any secret-key system), a single key is used for both encryption and decryption. Thus, this key must be transmitted through a communication channel, because the same key is used to encrypt the message by the sender, and decrypt this message by the receiver. Because a single key is used for both functions, this system is called “symmetric cryptography”. The transmission of this key raises the risk of vulnerability and increases the possibility of revealing it. However, in the PKI approach, private-keys never need to be transmitted or shared with anyone.

As mentioned in Section 2.5.2, the PKI approach can be used to sign information through various digital signature algorithms in order to ensure its integrity and confirm the identity of the signer (i.e., authentication of sender and receiver). This is not the case with any secret-key system, where the authentication process requires the sharing of some secret or selecting a trust third. This raises the suspicion that at anytime, any party may refuse the authentic message by claiming that the secret was somehow compromised. However, this problem is promoted by the PKI approach, and it can avoid this type of repudiation. This problem is promoted because the PKI links senders to their messages. Senders sign messages with their private key and therefore, all messages signed with the sender's private key originated with that specific individual. The reason for this is that each user has sole responsibility to protect his own private-key and not share it with any other party. Hence, PKI ensures that an author cannot refute that they signed or encrypted a particular message once it has been sent, which explains why this mechanism is called "non-repudiation".

There are two threats to the PKI approach; both of them are not considered disadvantages for this approach, because both of them can be avoided by using some precautions. The first threat PKI may encounter is the possibility to attack the certificate authority's key pair. This can be avoided by using long keys, and the CA should change them regularly. The second threat is if someone pretends to be someone else in order to obtain a certificate from the CA. For example, let us say Alice generates a public-key pairs and sends a request to the CA using the public-key to generate a digital certificate. At this time, if the CA is fooled and sends her said certificate, Alice can access Bob's information, because the CA issued Bob's certificate is the same party who issued Alice's certificate. The precaution that should be taken by the CA is to verify that the certificate request did indeed come from its purported requester (i.e., asking some confidential questions or stating some identification requirements and policies).

In the real world, certificate requests are sent from requesters to the CA by generating the key pairs. This thesis assumes that each role player sends his public-key to the neutral side to issue a certificate from the CA. These steps are encapsulated together using the OpenSSL tool (i.e., this will be explained in detail in Chapter 6).

3.6 Proposed framework

The above sections (Section 3.2, 3.3, 3.4, and 3.5) discussed all research problems and the research methodology for each problem. They discussed several points, including: how the tangible resources can be represented to accommodate the digital technologies using the LDP; how to foster trustworthiness among role players and judges by adding provenance metadata to the forensic information; how the lack of technical knowledge on digital evidence can be compensated by different consumption patterns of the semantic web; and how the represented information can be secured and shared on a closed-scale using PKI.

The proposed framework should reflect all these points. The use case diagram in Figure 3.7 shows and summarizes the main tasks that should be supported by the proposed framework, as well as the actors of each use case.

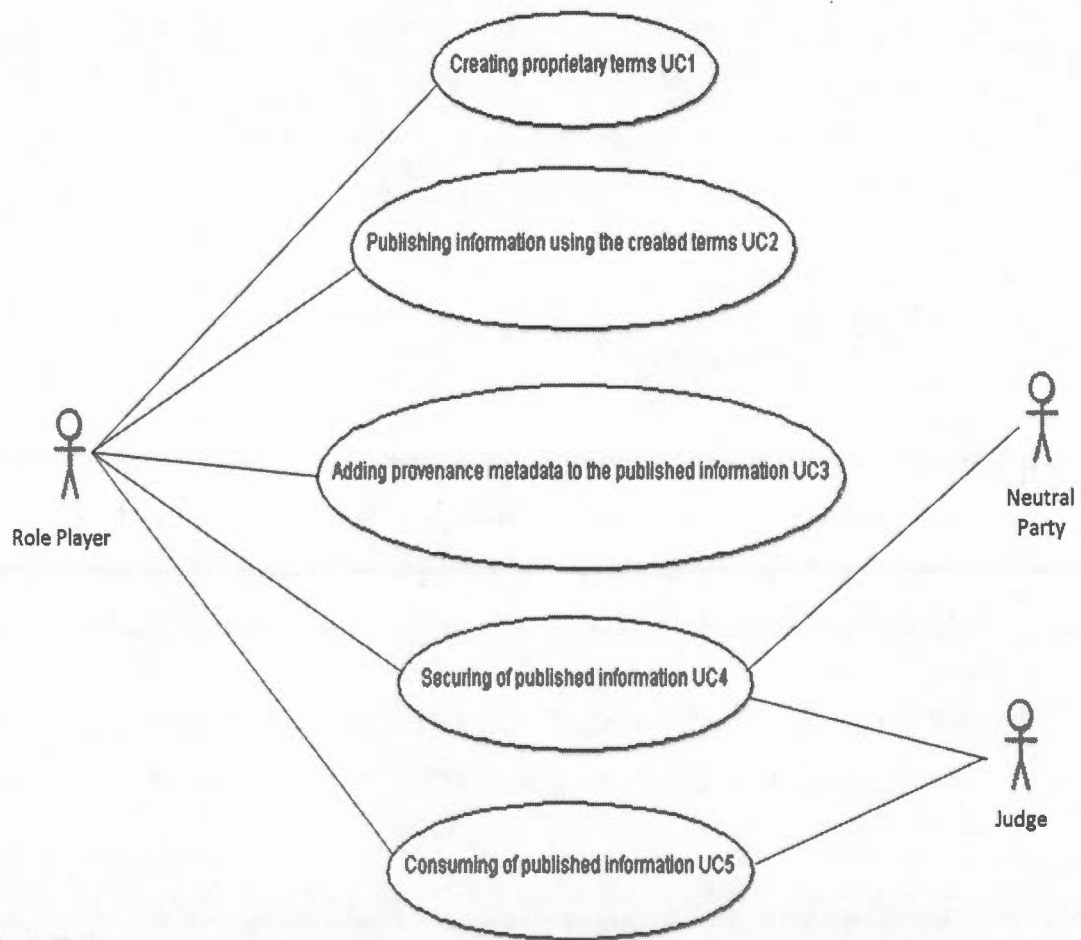


Figure 3.7 Use cases diagram of CF-CoC

The CF-CoC framework presented in Figure 3.8 consists of several modules. Each module is responsible to perform a set of tasks. The number assigned to each module is just for numeration (e.g., the PKI module is number six). The order presented in this figure starts by creating new proprietary terms (Module 2) using the vocabularies of the semantic web (Module 1), and they are annotated using provenance metadata (Module 4). Once those terms are created, they can be used to publish different forensic triples (Module 3). These triples can also be annotated using provenance metadata (Module 4). After publishing the forensic triples, they can be consumed

using different consumption patterns (Module 5). The Public-key Infrastructure module is used to publish and consume the information on a closed scale.

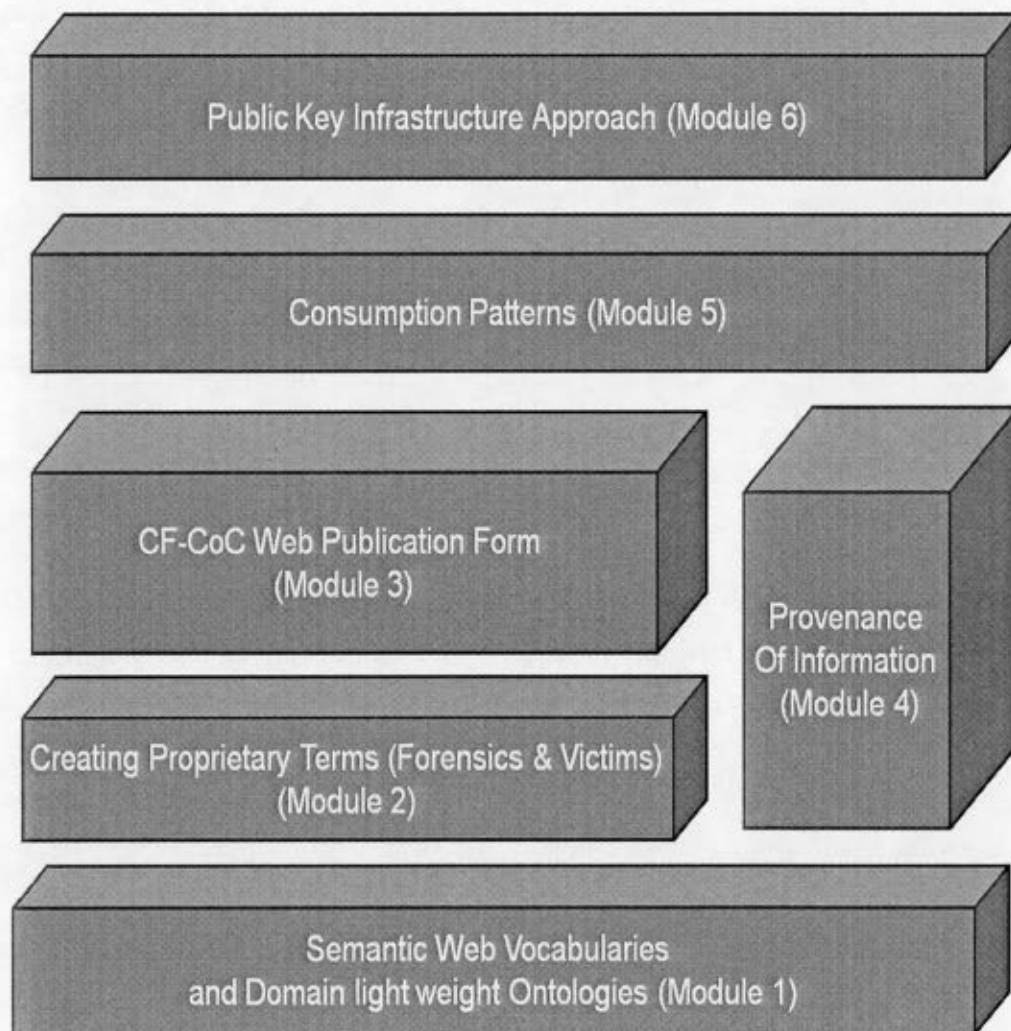


Figure 3.8 CF-CoC framework (Gayed, Lounis et al. 2013b)

As shown in Figure 3.8, the CF-CoC framework contains six modules. Each module reflects a solution for a problem. Table 3.2 indicates problem the module's framework resolves.

Table 3.2 Research problems and corresponding solutions

Problem	Module / Solution
Accommodation with digital technologies	Creating proprietary terms and publishing RDF statements (Module 1,2 and 3)
Fostering trustworthiness among role players and juries	Provenance metadata (Module 4)
Juries awareness about digital evidence	Consumption Patterns (Module 5)
Security of CoCs information	PKI Certificates (Module 6)

The solution of the first problem is presented in the first three modules of the framework. These modules are responsible for creating and defining all proprietary terms related to the victim and forensic parts with aid from the well-defined vocabularies of the semantic web.

The solution of the second problem is presented in the Module 4 of the framework, and this module adds different provenance metadata during the creation of terms, as well as during the use of such terms, to publish and describe the forensic information.

The third problem will be resolved through the consumption patterns module, and this is the fifth module of the framework. Different tasks will be implemented inside this module, such as browsing and serialization, crawling, reasoning, and querying.

The last problem will be resolved through the sixth module of the framework. This module is responsible for transforming the LOD to LCD by restricting the access to different resources among the role players and judges.

3.7 The framework environment

The CF-CoC framework is implemented using the “*Personal Home Page*” (Php) and “*easyRDF*”²³, “*Graphiz*” tool, and its graph objects are used within the “*easyRDF*” to produce and draw different RDF models. In addition, the operating system used is Windows, along with the Internet Information Services (IIS)²⁴ and the OpenSSL tool²⁵. IIS simulate the machine as a server, and the OpenSSL tool, which is widely used in implementing the Transport Layer Security (TLS), is used to create the digital certificates. Figure 3.9 shows the user interface of the CF-CoC implemented system.

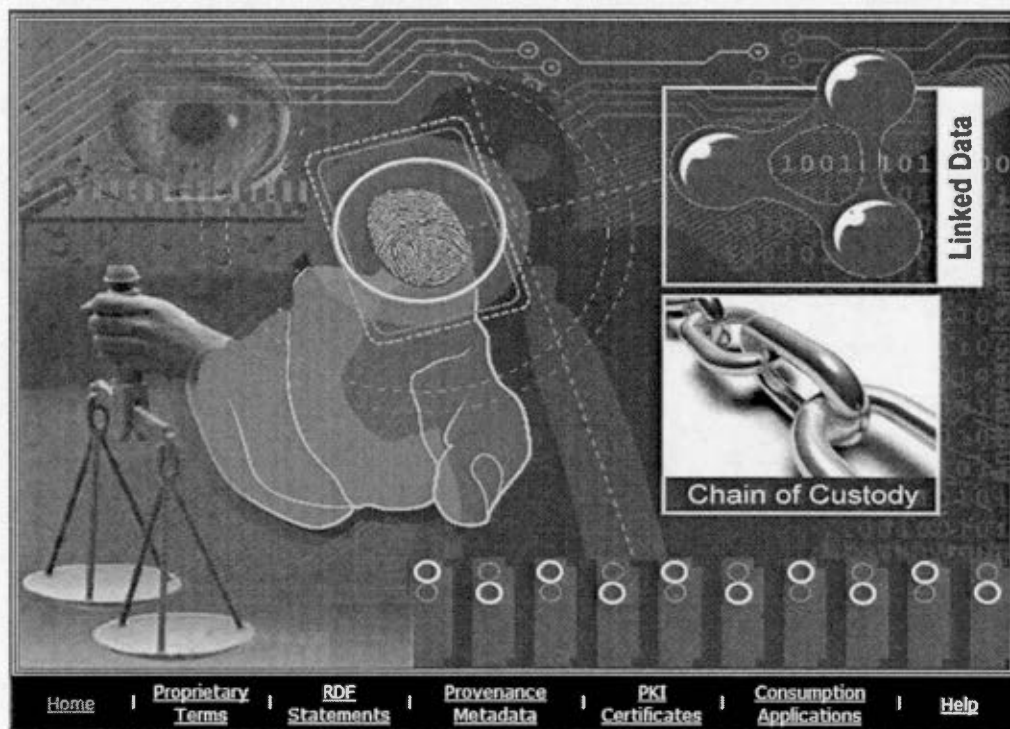


Figure 3.9 User interface of CF-CoC system

²³ <http://www.easyrdf.org>

²⁴ <http://www.iis.net>

²⁵ <https://www.openssl.org/>

3.8 Conclusion

This chapter discussed the research methodology for the research problems. It presents different facts to prove the four hypotheses proposed in Chapter 1. It started by discussing the several advantages of LDP and how such advantages can be exploited to serve the forensic information. It discussed how the named graph with provenance metadata can be applied to the forensic information to foster trustworthiness among role players and judges. In addition, it noted that the different consumption patterns of LD can be bended to help the judges to consume and understand the represented information. Furthermore, it discussed how digital certificates can be adapted to consume the published information from open scale to closed scale while keeping the resolvability advantages of resources. The aim of this adaptation was for consuming forensic information only among the role players and judges. Finally, it ended by a proposed framework that conciliates all research problems through different solution modules.

Therefore, the next three chapters will discuss the design and implementation of all modules in the framework. Chapter 4 depicts the first three modules that implement and explain the process of selecting, defining and publishing proprietary terms to represent the tangible CoC information. Chapter 4 will implement the annotation of different metadata throughout this process. Chapter 5 will discuss and implement the different consumption patterns used to consume the represented information. Lastly, Chapter 6 will explain how digital certificates can be issued by the Certificate Authority (CA) to restrict access to represented resources.

CHAPTER IV

CREATING AND PUBLISHING PROPRIETARY TERMS USING LIGHTWEIGHT ONTOLOGY AND ANNOTATING THEM USING PROVENANCE METADATA

4.1 Introduction

This chapter discusses how to create and publish proprietary terms using the RDF++ constructors and how to annotate them using provenance vocabularies of the semantic web. It mainly discusses the first three use cases presented in Figure 3.7. Figure 4.1 depicts the activity diagram of creating and annotating proprietary terms

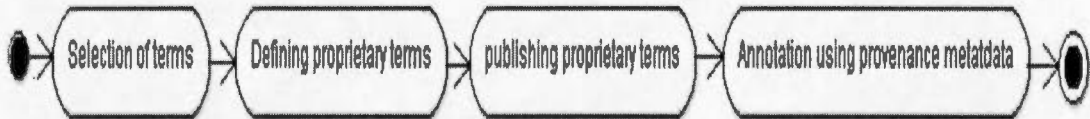


Figure 4.1 Activity diagram of creating and annotating proprietary terms

It starts by determining the forensic terms from the tangible CoC. Each technician participating in a forensic investigation process should select and determine the forensic terms describing his own chain of custody. This task is subjective and intellectual. This means that it depends on the perception of the role player to describe and select the proper terms that can describe his forensic information.

Before the role player starts to determine the forensic terms, he may search for other terms on other ontologies created by other role players to describe different or same

concepts/objects of the world. In case of redundancy of terms describing the same object, this will not affect the quality of LD due to the two reasons mentioned in Section 2.2.1.5: terms can be dereferenceable and mapped.

The T-Box phase: after determining the terms, the role player starts to define his own proprietary terms over three steps. First, he should create the container of his terms (i.e., to which phase the terms will belong). Then, he creates the category of the terms (i.e., to which task the terms will belong). Finally, he starts to define the terms using the different constructors of RDF++.

The A-Box phase: after creating the proprietary terms, the role player starts to use them to publish and describe his chain of custody, together with the aid of different well-defined vocabularies of the semantic web.

Along the definition and publication of terms, the role player can inject different provenance metadata vocabularies to annotate the forensic information during the T-Box and A-Box phases. For instance, the current framework will use the most popular provenance vocabularies, such as DC and FOAF.

In the following sections, an example for a tangible CoC will be used. This CoC is owned by a certain technician and contains some information related to a forensic phase. This chapter depicts how the technician can represent and transform this tangible document to an *e*-CoC by the aid of the CF-CoC framework.

Let's consider the following case study example for a CoC. It presents the preservation task retrieved from the acquisition phase of the Kruse model:

“The name of the first responder in the acquisition phase is Jean-Pierre. He is the role player of this phase, and he preserved the state of the digital media, PDA device, which has the SN: 0G-4023-32-362. The date he did this task is March 5th 2014” (Gayed et al., 2013b, 2014b, 2015).

Before the role player starts his work to select and define the terms from this CoC, an exchange of digital certificates must take place between CA, neutral side that hosts the CF-CoC system, judge and role players. The CA should issue digital certificates for role players and judge to publish and consume in a secure manner the forensic information on the web of data.

The above tangible CoC will also be considered along the next two chapters to illustrate how the represented information can be consumed using different consumption patterns (Chapter 5) and in a secure way (Chapter 6). This example is also structured in a way that tries to encompass all lightweight ontology constructors.

4.2 Selection of terms

The tangible CoC provided in last section is retrieved from the Kruse model. It describes some information from its preservation forensic phase.

The first step to create an *e*-CoC from this tangible CoC is to identify the terms (see Table 4.1) (i.e., as we mentioned in Section 4.1, identifying proprietary terms are a subjective task and may differ from the perception of one creator to another).

This case study contains T-Box and A-Box information. Terms of T-Box are of type class and property. Also it contains the term name of forensic phase, which will be of type Ontology object. This case contains some terms that should be defined to describe instance of data. For example, the “*FirstResponder*” term defined in table 4.1 are defined using known vocabulary of the semantic web to instantiate members and publish triples. This is also the same case for a term like the “*RolePlayer*”.

Table 4.1 Proprietary terms of preservation task

Box Type	Term name	Term Type
T-Box	FirstResponder	Class
	RolePlayer	Class
	Acquisition	Ontology Object
	DigitalMedia	Class
	preserve / preservedBy	Property
	SN	Property
A-Box	Jean-Pierre	Resource/Instance
	PDA device	Resource/Instance
	0G-4023-32-362	Literal String (Plain/Typed)

The “*DigitalMedia*” is also another term of type class that can be instantiated to describe different data instances, such as different media devices (i.e., hard disk, thumb drive, digital camera, etc.)

For the properties, Table 4.1 shows some property terms like the “*preserve*” term. Simply the terms of type property are the terms that can establish a relation between the subject and object in RDF model. For example in this case, the preserve term can relate the player who did the preservation task with the preserved digital media.

A property term can be explained by other points of view. It may be presented in the form of passive voice (e.g., “*copy*” and “*copiedBy*”), or it may describe or add supplementary information to an object (e.g., the age of the role player is 58 years old and it is a predicate for adding supplementary information about the role player).

4.3 Defining proprietary terms

As mentioned in Section 4.1, this task will be performed over three steps. Firstly, the role player defines the ontology, then the category, and finally, the terms. Before this process starts, the role player should have his own client certificate and a CA public certificate installed on his machine to access the CF-CoC system, provided to him from the CA authority. This will be explained in detail in Chapter 6.

4.3.1 Creation of ontology object (vocabulary)

The task of creating ontologies is about to create the ontology object or the vocabulary of the acquisition phase (see Figure 3.4). The domain name field is required to mint the ontology to a unique domain name owned by a neutral part (i.e., second aspect in Section 2.2.1.5). The screen for creating the ontology object is shown in Figure 4.2. For simplicity, the domain name shown below is the local IP where the CF-CoC system is residing.

The screenshot shows a web application titled "DIGITAL FORENSICS" with a navigation bar containing links: Home, Proprietary Terms, RDF Statements, Provenance Metadata, PKI Certificates, Consumption Applications, and Help. The main content area is titled "Create Forensic Phase" and contains a form with the following fields:

Object Type :	Ontology		
Root Folder :	vocab/		
Domain Name : *	<input type="text" value="https://127.0.0.1"/>		
Publisher / Role Player :	Select your certificate: *	<input type="text" value="Jean-Pierre"/>	<input type="button" value="Browse..."/> <input type="button" value="Submit Cert"/>
	Value Type	<input type="text" value="Resource"/>	
Ontology/Phase Name : *	<input type="text" value="Acquisition"/>	(Specify the CoC phase name / e.g. acquisition)	
Label : *	<input type="text" value="Acquisition Phase vocabulary"/>	(Specify the phase description / e.g. acquisition)	
Creation / Publishing Date : *	<input type="text" value="5 March 2014"/>		
<input type="button" value="Create New Ontology"/>			

Figure 4.2 Screen for creating an ontology

Figure 3.4 shows that ontology is corresponding to a forensic phase. As shown on Figure 4.1, this task is about creating a forensic phase of type ontology. Usually, a domain name is composed of a string of characters that can also be substituted by its corresponding Internet Protocol (IP) address.

The role player should also submit his own digital certificate provided to him and select if this resource will be a terminal resource or a non terminal resource. If it is a terminal resource, the role player can add extra information about himself using different terms (custom or build-in terms).

The name and label of the forensic phase should be defined to identify this ontology. Also some provenance metadata may be added to mention the creation date of this ontology. Some other fields may be added to this module to give the opportunity to the role player to describe more information about this container (i.e.,

Ontology/forensic phase). Once all the necessary information is completed, the role player confirms the creation of the forensic phase as shown in Figure 4.3.

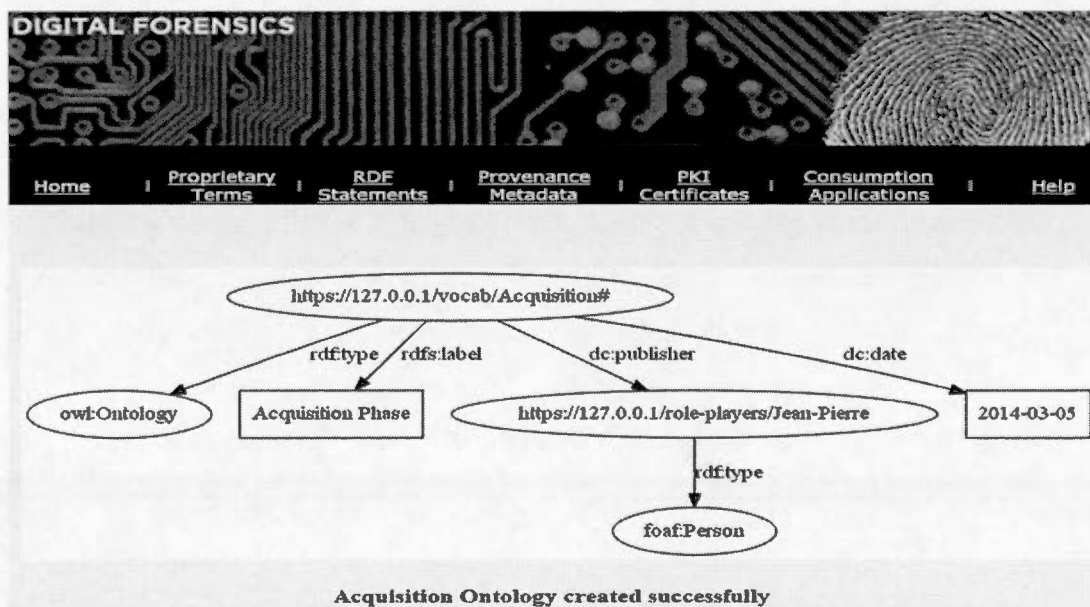


Figure 4.3 RDF screen model of the acquisition ontology

This graph and all the following RDF figures are generated by using the “*Graphviz*” module²⁶, which is integrated within the CF-CoC system.

After creating the acquisition ontology, the role player proceeds to the task module to create terms and append them to this new ontology object.

Before going further, an important note should be mentioned on creating proprietary terms: it is about the URI used to identify the ontology. This is a URI of type hash, and any new term will be appended to this string of characters (e.g., if for example a new term X is defined, then this term X will be appended to the suffix position of the mentioned URI). For simplicity, the abstract URI of the domain (i.e.,

²⁶ <http://www.graphviz.org>

<https://127.0.0.1/vocab>) will be replaced by its name space (i.e., “*cf-coc:*”) proposed by the CF-CoC system.

4.3.2 Creation of new terms

This task relates to four essential entries. The first entry is the term name. The second entry is selecting to which ontology the role player will append his new proprietary term. The third entry specifies the category/forensic task (see Figure 3.4 and Figure 4.4). The category could be one of the three tasks provided in Section 2.2.2.1 (preservation, recovery, or copy). In this field, the user may select ‘New’ to create a new category or select ‘Existing’ to import an existing category defined in another vocabulary (ontology), created by another role player (i.e., two different forensic phases may have common category/task). The last field is the selection of term type (i.e., a term can be a property or a class).

DIGITAL FORENSICS

Home | Proprietary Terms | RFE Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Create New Forensic Term

Term Name : *	<input type="text"/> (Specify the name of the new term)
In Ontology : *	<input type="text" value="- Please Select -"/> (Specify in which ontology you define a new term)
Category : *	<input type="radio"/> New <input type="radio"/> Existing
Term Type : *	<input type="text" value="- Please Select Type -"/> (Specify the type of the new term)

Figure 4.4 Screen for creating a proprietary term

Referring to Table 4.1, there exist seven terminological terms that should be defined (i.e., T-Box row). There exist two types of ontologies in the CF-CoC system, the custom ontologies created by different role players, and the built-in ontologies created for the semantic web. A role player cannot append a proprietary terms to built-in vocabularies; he can append them to custom vocabularies.

The build-in ontologies, called also well-known vocabularies, are those that already exist on the semantic web. Their terms should be reused to describe data wherever possible, rather than reinvented (e.g., Friend of a Friend and Dublin Core).

4.3.2.1 Class terms

Table 4.1 contains three terms of class type. This section discusses how such terms are defined. In this section, the root definition of any term of class type is a class from a well-defined vocabulary of the semantic web.

- The “*RolePlayer*” term:

The “*RolePlayer*” will be defined as a term of class type and a subclass of the class Person of the FOAF (friend of a friend) ontology (McGuinness et Van Harmelen, 2004; Brickley et Miller, 2014). In addition, the “*RolePlayer*” term will belong to a forensic task called “*Preservation*” and it is a new category (i.e., forensic task) in this forensic phase (i.e., “*Acquisition*”). Label and comment can be added to identify, give a hint about the term, or why it is created by the role player.

DIGITAL FORENSICS

Home | Proprietary Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Create New Forensic Term

Term Name : *	RolePlayer (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input checked="" type="radio"/> New <input type="radio"/> Existing	
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Class Name	Person (FOAF Basics)
<input checked="" type="checkbox"/> Label	Enter a label for the term RolePlayer	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term Each phase has a role pla	
Create New Term		

Figure 4.5 Screen for creating the “*RolePlayer*” class

- The “*FirstResponder*” term:

The “*FirstResponder*” term is a term that describes a specific role played by the role player. As has been mentioned, the role player may be an officer; investigator, expert witness, prosecutor, defender, etc (see Section 1.1). Thus, the “*FirstResponder*” is a sub-class of the predefined “*RolePlayer*” term. This term, also, will be appended to the acquisition container and belongs to the predefined category “*Preservation*”, which belongs to the same ontology (see Figure 4.6).

DIGITAL FORENSICS

Home | Proprietary Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Create New Forensic Term

Term Name : *	FirstResponder (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input checked="" type="radio"/> New <input type="radio"/> Existing	
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	RolePlayer (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term FirstResponder	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term player of Acquisition phase	
Create New Term		

Figure 4.6 Screen for creating the “FirstResponder” class

As noticed, during the definition of these two classes, the most important constructors related to this definition is the sub class constructor of the RDFS vocabulary (i.e., “*rdfs:subClassOf*”).

Referring to Section 2.2.1.2, if the player of the preservation task in the acquisition phase is called “Pierre” and he is of type “FirstResponder” T(Pierre, *rdf:type*, FirstResponder), this implies that he is also of type “RolePlayer” T(Pierre, *rdf:type*, RolePlayer), because the “FirstResponder” is of type class and it is a subclass of the class “RolePlayer”.

- The “DigitalMedia” term:

This term can be used to describe any type of media device used in the forensic process. This term can be defined as a subclass of “*owl:Thing*”. The latter is considered in the OWL vocabulary as a root of the overall taxonomy

of resources, and every individual in the OWL world is a member of this class (see Figure 4.7).

DIGITAL FORENSICS

Home | Proprietary Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Create New Forensic Term

Term Name : *	DigitalMedia (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Acquisition Preservation
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From	Built-in Ontology
	Ontology Name	Ontology_Web_Language (owl)
	Class Name	Things (Owl Semantics)
<input checked="" type="checkbox"/> Label	Enter a label for the term Digital Media	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term DM Device Media	
Create New Term		

Figure 4.7 Screen for creating the “*DigitalMedia*” class

4.3.2.2 Property terms

Table 4.1 contains three terms of property type. This section discusses how such terms are defined. In this section, the root definition of any term of property type is a class from a well-defined vocabulary of the semantic web. In case of defining property terms, the domain of subject and the range of objects should be determined. Other OWL vocabularies can be used to enrich the property terms with lightweight

axioms such as “*owl:inverseOf*”, “*owl:FunctionalProperty*”, and “*InverseFunctionalProperty*” (i.e., will be explained in Chapter 5).

As has been mentioned, the terms of property type are terms that can relate subjects with objects, add supplementary information between subjects and objects (i.e., lightweight axioms), or provide inverse relation with another predicate. There are two important constructors when defining terms of property type: the “*owl:ObjectProperty*” and “*rdfs:subPropertyOf*”. The former is used to indicate that the term defined is of type property and the latter that the defined term has a relationship with another property.

- The “*preserve*” term:

The “*preserve*” term is a task (i.e., verb, action) meaning that someone preserves something. In this case, the first responder can preserve the status of a digital media at first hand of his forensic task. Thus, the domain and range of this term can be easily defined by the role player to record and describe “*who*” preserved “*what*”. Simply said, the domain will be a class of type “*foaf:Person*”, and the range will be a class of type “*owl:Thing*”.

According to the terms defined above, the domain can be the “*FirstResponder*” class, and the range can be the “*DigitalMedia*” class. Domain and range values can also have type “*foaf:Person*” and “*owl:Thing*”, respectively.

As well, the “*preserve*” term can inherit from a well-defined vocabulary term called *foaf:made*. According to the definition of this constructor in (i.e., http://xmlns.com/foaf/spec/#term_made), it defines something that is made by an agent. This means it relates an agent to something.

Figure 4.8 explains how this term can be defined, and shows that an extra constructor is also selected. This constructor is the “*owl:InverseFunctionalProperty*” (i.e., this will be explained in Chapter 5

with another constructor “*FunctionalProperty*” to discuss another use case example in Chapter 5). The same idea can be applied to the passive voice of the term (i.e., “*preservedBy*”).

Property Name		made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	DigitalMedia (Preservation)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	FirstResponder (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term: preserve	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term: FR preserve DM	
OWL Vocabulary		
<input checked="" type="checkbox"/> Inverse-of	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Property Name	preservedBy (Preservation)
<input type="checkbox"/> Functional Property		
<input checked="" type="checkbox"/> Inverse Functional Property		

Figure 4.8 Screen for creating the “*preserve*” property

As shown in Figure 4.7, the range and domain of the “*preserve*” term are terms selected from the custom ontology created and discussed in Section 4.3.1.

- The “*preservedBy*” term:

Same explanations given for the “*preserve*” term can be applied to the “*preservedBy*” term. It has the opposite meaning of “*preserve*” term. The *owl:inverseof* is used with “*preservedBy*” to show the inverse with the “*preserve*”. If the “*preserve*” predicate is tagged to be “*inverseFunctionalProperty*” then the “*preservedBy*” is tagged to be

“*FunctionalProperty*”. Also, the domain and range will also be inverted. In such case, when the first responder preserves a digital device, it is also correct to say that the digital device is preserved by the first responder. This means that the domain of “*preservedBy*” term will be “*DigitalMedia*” class and its range will be “*FirstResponder*” class (see Figure 4.9).

term type: ~

property (Specify the type of the new term)

RDF-Schema Vocabulary

☒ Subproperty-of

From: Built-in Ontology

Ontology Name: Friend_of_a_Friend (foaf)

Property Name: made (Documents and Images)

☒ Range

From: Custom Ontology

Ontology Name: Acquisition (cf-coc-Acq)

Class Name: FirstResponder (Preservation)

☒ Domain

From: Custom Ontology

Ontology Name: Acquisition (cf-coc-Acq)

Class Name: DigitalMedia (Preservation)

☒ Label

Enter a label for the term: preserved by

☒ Comment

Enter a comment for the term: DM preservedBy DM

OWL Vocabulary

☐ Inverse-of

☒ Functional Property

Figure 4.9 Screen for creating the “*preservedBy*” property

The task of selecting which constructors can be used to tag property terms (e.g., “*preserve*”, “*preservedBy*”) is a subjective task as long as it is reflecting a feasible case. It may differ from one role player to another. For example, in the screens shown above, the “*preserve*” is tagged as “*InverseFunctionalProperty*”, this is a case when a role player, creator of the term, wants that each device is the object of an action (*preserve*) by a single role player. However, if the role player wants that each role player is the subject of an action that preserves one single device, this means that each role

player can preserve only one device. Chapter 5 gives an example of tagging the “*preserve*” property with “*FunctionalProperty*”.

- The “*SN*” term:

The last term of the T-Box mentioned in Table 4.1 is the serial number. This property can provide supplementary information about the subject of type “*owl:Thing*”. This means that an RDF triple containing the SN term can be provided to specify extra information to a media device.

In this case, the domain of the “*SN*” term will be the subject of type “*owl:Thing*” and the range will be a string of characters that identify this device of type “*rdfs:Literal*”. “*SN*” is also a sub-property of the identifier predicate of the DC (*dc:identifier*).

This type of term may also be associated to another constructor such as “*owl:InverseFunctionalProperty*”. This constructor adds an axiom to the property term. The reason of using this constructor with this term will be explained in Chapter 5. Figure 4.10 shows how the “*SN*” term is defined.

Property Name		Identifier (http://purl.org/dc/terms/identifier)
<input checked="" type="checkbox"/> Range	From	Built-in Ontology
	Ontology Name	Resource Description Framework Schema (rdfs)
	Class Name	Literal (Literal Values of String and Integers)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	DigitalMedia (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term: Serial Number	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term: SN identifies DM	
OWL Vocabulary		
<input checked="" type="checkbox"/> Inverse-of	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Property Name	preservedBy (Preservation)
<input type="checkbox"/> Functional Property		
<input checked="" type="checkbox"/> Inverse Functional		

Figure 4.10 Screen for creating the “*SN*” property

The next two figures (Figure 4.11 and 4.12) illustrate the T-Box ontology for the “SN” and “preserve” predicates:

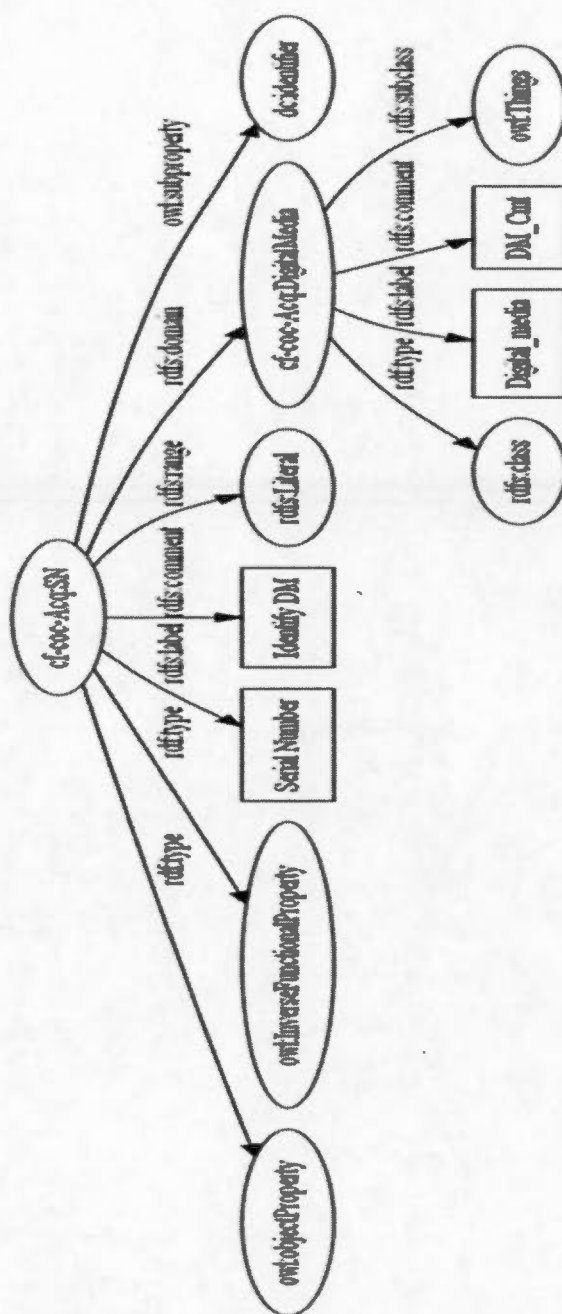


Figure 4.11 T-Box of “SN” property

As has been noticed, during the definition of terms, some provenance vocabularies are used such as FOAF and DC. The usage of such vocabularies refers to the possibility to inject different provenance terms during the creation of proprietary terms. The terms provided are not that much used to annotate and add supplementary information to the forensic terms being defined as they are used to define the terms themselves.

4.4 Publication of proprietary terms

Module three is a straightforward module. All custom terms that have been defined in the proprietary terms module (T-Box) can be used to publish and describe the CoC in form of RDF triples. Not only custom terms are used to publish RDF statements, but also terms from the well-known vocabularies can be used to publish such RDF statements. Next figure shows the class diagram of an *e*-CoC.

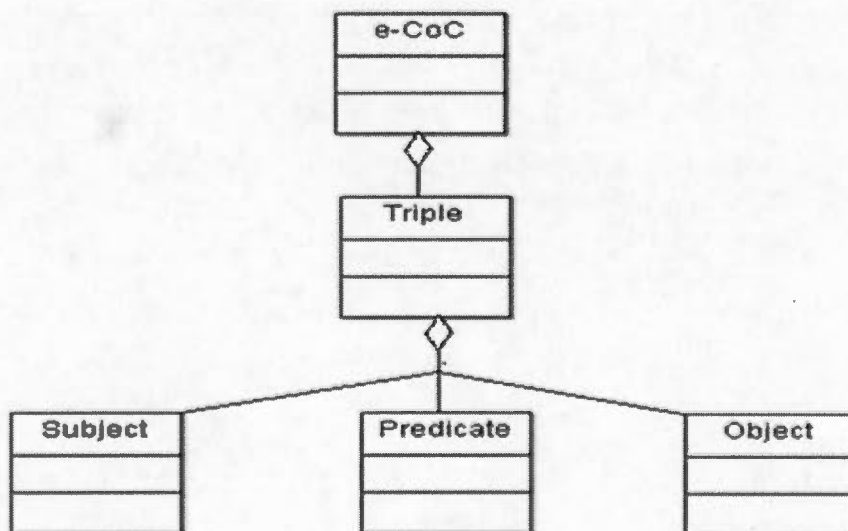


Figure 4.13 Class diagram of *e*-CoC

The main tasks in this module are the publication of terms. Publication of terms is about selecting the subject, predicate (property), and object. For mapping between terms, different constructors from OWL vocabulary can be used such as “*equivalentProperty*”, “*equivalentClass*”, and “*sameAs*” (see Table 2.2).

The property slot of the triple defines the object values of the subject. On its left (subject), we define the domain, and on its right (object) we define the range (see Table 2.1), also subclasses are defined.

For instance, the property term “*preserve*” defined in the T-Box, has “*FirstResponder*” class (subject) as domain, and “*DigitalMedia*” class (object) as range. Thus, any resource selected/created by the publisher in the subject slot of RDF triple will be of type “*FirstResponder*”, which is a subclass of “*RolePlayer*”, which is a subclass of class “*Person*” defined in the FOAF vocabulary; see Figure 4.12.

As shown in Figure 4.14, there are three slots. The predicate slot shows that role players can select the property from one of two types of ontologies, whether from built-in ontologies, or from custom ontologies.

The selection of a predicate is the primary selection to construct the RDF triple. After that, the role player defines the subject and object of the triple.

DIGITAL FORENSICS

Home | Property Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Publish RDF Triples

Subject	Predicate	Object
<input type="radio"/> New Resource <input type="radio"/> Existing Resource	From <input type="text" value="-Select Ontology Type-"/> <input type="text" value="-Select Ontology Type-"/> <input type="text" value="Built-in Ontology"/> <input type="text" value="Custom Ontology"/>	<input type="radio"/> New Resource/Literal <input type="radio"/> Existing Resource/Literal
<input type="button" value="Publish and Draw"/>		

Figure 4.14 Screen for publishing RDF triples

As mentioned, the predicate has already its domain and range published within the definition of the term. The system automatically aids the role players to know/remind him about both values of the domain and range to facilitate him the choice of resources for subject and resources/literals for object he can propose to publish his RDF triple.

For example if the role player wants to select the predicate of “*SN*”, he starts to select from the predicate slot the ontology type (i.e., custom), then from the ontology name, he selects the ontology to which the “*SN*” term belongs (i.e., acquisition).

Once the predicate slot is selected, the system displays the domain (subject) and range (object) slots. This will be very useful for role players who want to use terms defined by other role players, on other ontologies, to properly publish and understand the role and aim of such terms (See Figure 4.15).

Publish RDF Triples

Subject	Predicate	Object
<input type="radio"/> New Resource <input type="radio"/> Existing Resource	From <input type="text" value="Custom Ontology"/> Ontology Name <input type="text" value="Acquisition (cf-coc-Acq)"/> Property Name <input type="text" value="SN (Preservation)"/>	<input type="radio"/> New Resource/Literal <input type="radio"/> Existing Resource/Literal
Domain : DigitalMedia		Range : Literal

Figure 4.15 Screen showing domain and range of a proprietary term

After the role player selects the predicate, he starts to select the subject and object. A role player can select an existing URI resource or define new resource to the custom ontology. If he selects to create new resource, the only ontology that will appear to him is the one that he created after he installed the digital certificates. However, if the role player selects an existing resource, he can select an existing resource from stored in the system. This means that a role player can not publish a resource in another forensic phase, but he is able to use an existing resource published by another role player in another forensic phase. Also, the system guides the role players to identify the type of resources/individuals through the range and domain of the selected predicate.

The second main task of this module is mapping between terms. The predicate slot of this triple will be one of the three constructors mentioned in Chapter 2 (i.e., “*owl:equivalentProperty*”, “*owl:equivalentClass*”, and “*owl:sameAs*”, see Table 2.2). The “*owl:equivalentProperty*”, “*owl:equivalentClass*” are used on the level of T-Box. However, “*owl:sameAs*” can be used on both levels to map between properties and classes on the level of T-Box, and between individuals on the level of A-Box. Only on OWL Full, where a class can be treated as instances (a class can be

treated simultaneously as a collection of individuals and as an individual in its own right), we can use the *owl:sameAs* to define class equality and indicating two concepts have the same intentional meaning.

For “*owl:equivalentClass*”, if the role player uses this constructor, then he is going to map between two classes (does not mean same identity, but same class extensions, this means that both classes contain exactly the same set of individuals).

If a role player uses the “*owl:equivalentProperty*”, then he is going to map between two properties (does not mean same identity, but same property extensions, this means that both properties have the same values).

As explained in chapter two, it is unrealistic to assume everybody will use the same name to refer to individuals. That would require some grand design, which is contrary to the spirit of the web. The said mapping constructors (classes, properties, and individuals) will be very useful within the CF-CoC system. They will not be only used to map various terms, defined by different role players, but they can be exploited to automate the mapping process instead relying on role players to detect equivalent/equated terms. This can be easily achieved. For equivalence between class terms, the system can compare between classes to identify those that have same set of individuals. For equivalence between property terms, the system can compare among between properties to identify those that have same values. For equating between classes, properties, and individuals, the *owl:sameAs* can be used directly (case of OWL Full).

Figure 4.16 shows the *e*-CoC (A-Box) of the forensic preservation task. This generated ontology does not answer all the questions of CoC. It answers only the Who: Jean-Pierre, What: PDA device, and When: publication date of ontology. In order to have the answers to other questions, more terms need to be determined and defined. In this figure, the “*cf-coc-Acq*” is the prefix namespace of the acquisition

ontology. “*Jean-Pierre*” is an instance/resource from the “*FirstResponder*” class (i.e., which is a sub-class of the “*RolePlayer*” class), “*PDA device*” is an instance of “*DigitalMedia*” (i.e., which is sub-class from “*owl:Thing*” class. Any user-defined class is implicitly a subclass of *owl:Thing*), and “*preservedBy*” is the inverse property of the “*preserve*” property. “*SN*” is a “*FunctionalProperty*” where its domain is the “*PDA device*” and its range is “*0G4023-32-362*” (i.e., which is a string typed literal of *Literal* class). In addition, the forensic information coming from a forensic tool (e.g., AFF4), can also be integrated in the same CF-CoC framework (see Figure 3.3).

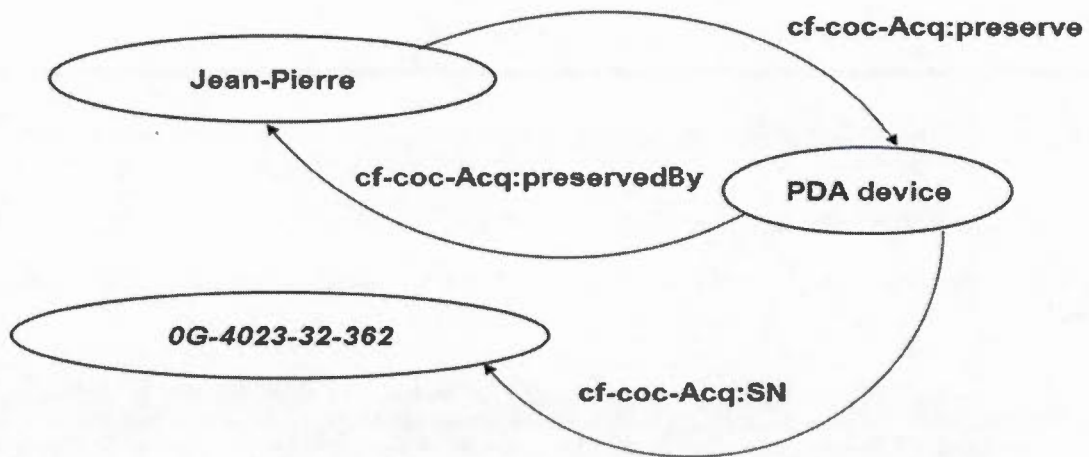


Figure 4.16 A-Box ontology of the forensic preservation task ($e\text{-CoC}_{\text{Acq}}$)

4.5 Annotation using provenance metadata

This section discusses the fourth module (UC2) presented in Figure 3.7. Along the definition and publication of terms, different provenance metadata can be added. The CF-CoC system uses the named graph method to add provenance metadata to a set of triples by naming them using URI (see Section 3.2).

An example of metadata added to the level of terms presented in Figure 4.16, where the DC vocabulary is used to answer when the ontology is published and who published it. Provenance metadata can be attached during the phase of T-Box and A-Box.

Figure 4.16 is a good example to add provenance metadata using the named graph method. This figure represents the *e*-CoC of the state preservation task in the acquisition phase. As explained in Figure 3.5, it provides abstract models for the named graph. The $NG_{\text{acquisition}}$ is the named graph of the acquisition phase, which contains three tasks. One of them is the preservation task provided in Figure 4.16.

Figure 4.17 depicts how provenance metadata are added to a named graph. The CF-CoC assigns automatically the URL address to each ontology by adding a suffix NG to the ontology URL. For example, if the URL of acquisition ontology is “https://127.0.0.1/Acquisition.rdf”, the URL of the acquisition ontology will be “https://127.0.0.1/AcquisitionNG.rdf”. In the same screen, the CF-CoC requires to select the ontology from which the role player can select the desired property from different provenance vocabularies (e.g., DC, FOAF).

Home	Property Terms	RDF Statements	Provenance Metadata	PKI Certificates	Consumption Applications	Help
----------------------	--------------------------------	--------------------------------	-------------------------------------	----------------------------------	--	----------------------

Provenance Metadata to Named Graph

Ontology Type	Custom Ontology
Ontology Name	Acquisition
The Named Graph URL is :	https://127.0.0.1/AcquisitionNG.rdf
Add Provenance metadata to NG From	From Built-in Ontology
	Ontology Name Dublin Core
	Property abstract
Enter Literal :	

abstract
 accessRights
 accrualMethod
 accrualPeriodicity
 accrualPolicy
 alternative
 audience
 available
 bibliographicCitation
 conformsTo
 coverage
 created
 creator
 contributor
 dateAccepted
 date
 dateCopyrighted

Add Provenance Metadata

Cyber forensics : Representing Chains of Custody (data principles)

Figure 4.17 Screen for adding provenance metadata to the NG

This screen task will lead to an annotated graph similar to the one provided in Figure 2.9, but for the acquisition named graph.

4.6 Conclusion

This chapter explained how the CF-CoC system can create forensic proprietary terms. After creating such terms, this chapter discussed how to use such terms to publish forensic information and annotate them using provenance metadata.

The next chapter will discuss how the *e*-CoC will be consumed by judges and role players using different consumption patterns. These patterns will aid them to consume the published information and navigate between different represented resources to get more information and understand the case in hand.

CHAPTER V

CONSUMPTION PATTERNS

5.1 Introduction

This chapter discusses the fifth module of the CoC framework. As mentioned, the LD is a style of publishing information that makes them easy to interlink, discover, and consume represented resources. This is achieved by making URIs, which identify data items, dereferenceable into more RDF descriptions resources (see Section 2.2.1, 3.2.3 and 4.3.1). The four patterns that can be used by consumers of the web of data are: browsing, searching, querying, and reasoning. Figure 5.1 shows the use case of pattern consumption (UC5):

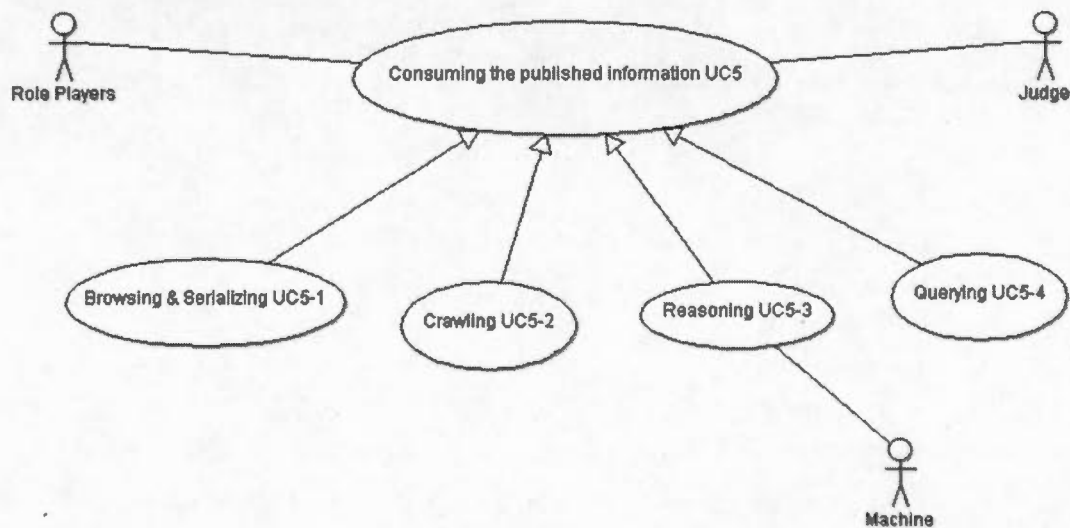


Figure 5.1 Use case diagram of consumption patterns

As mentioned, the consumers are the role players, judge, and machine. The role players and judge can use all patterns to consume the represented information. The reasoning pattern is the only pattern that can be used by judge and role players, but also can be used by the machine to infer implicit information.

The consumed information is the forensic information presented by the role players and inferred by the machine. After defining and publishing the forensic information (see Chapter 4), the role players and judge can consume the represented resources using such patterns. In addition to these patterns, an extra option has been added to the framework, with the browsing module, it is the generation of serialized code from RDF models using different serialization RDF languages (see Section 2.2.1).

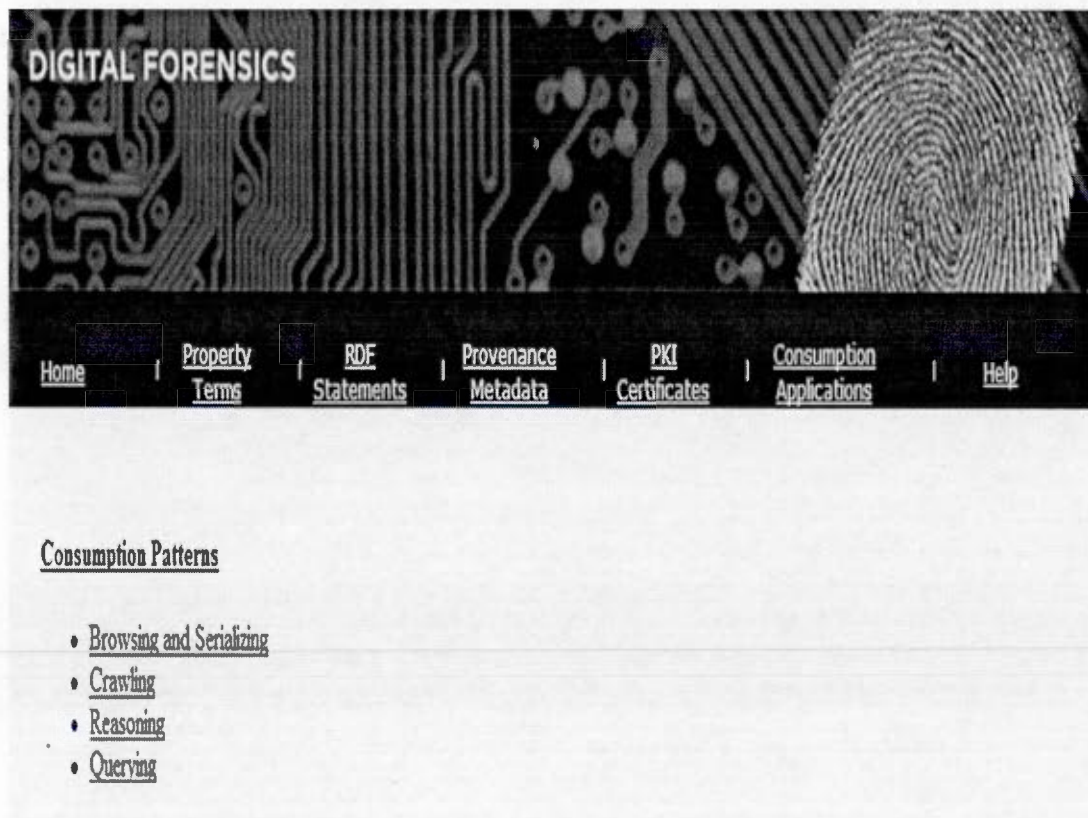


Figure 5.2 Screen for consumption patterns

The main objective of this chapter is to present how role players and judge can consume and understand the forensic resources. The used patterns must be adapted to their knowledge level. This chapter discusses how these patterns are used within the CF-CoC system in a way that can facilitate the consumption of such resources and exempt consumers (i.e., especially the judge) to know details related to each pattern. Each pattern is explained apart through the same example provided in Chapter 4. The above figure (Figure 5.2) shows the screen of different consumption patterns in the CF-CoC system.

5.2 Browsing and serializing

As mentioned in chapter 2, the browse pattern of LD is the same idea of browsing web documents. Both use the style of “follow-your-nose” to navigate between documents and resources. With this module, consumer will not need to know or learn about the different semantic browsers (see Section 2.4.1).

The browsing module implemented in this system allows the consumers to expand (i.e., dereference) the represented resources in order to get and understand the meaning of such resources. Consumer will also be able to see all the *e*-CoC published using these represented resources.

Figure 5.3 shows the screen for browsing. In this screen the CF-CoC system asks the consumers, on which ontology (i.e., forensic phase), they would like to browse resources. There are two main choices for consumers: whether to list all forensic phases published by all role players, or to select a specific phase to list all related resources. Once this choice is performed, the system displays the results of the selection.

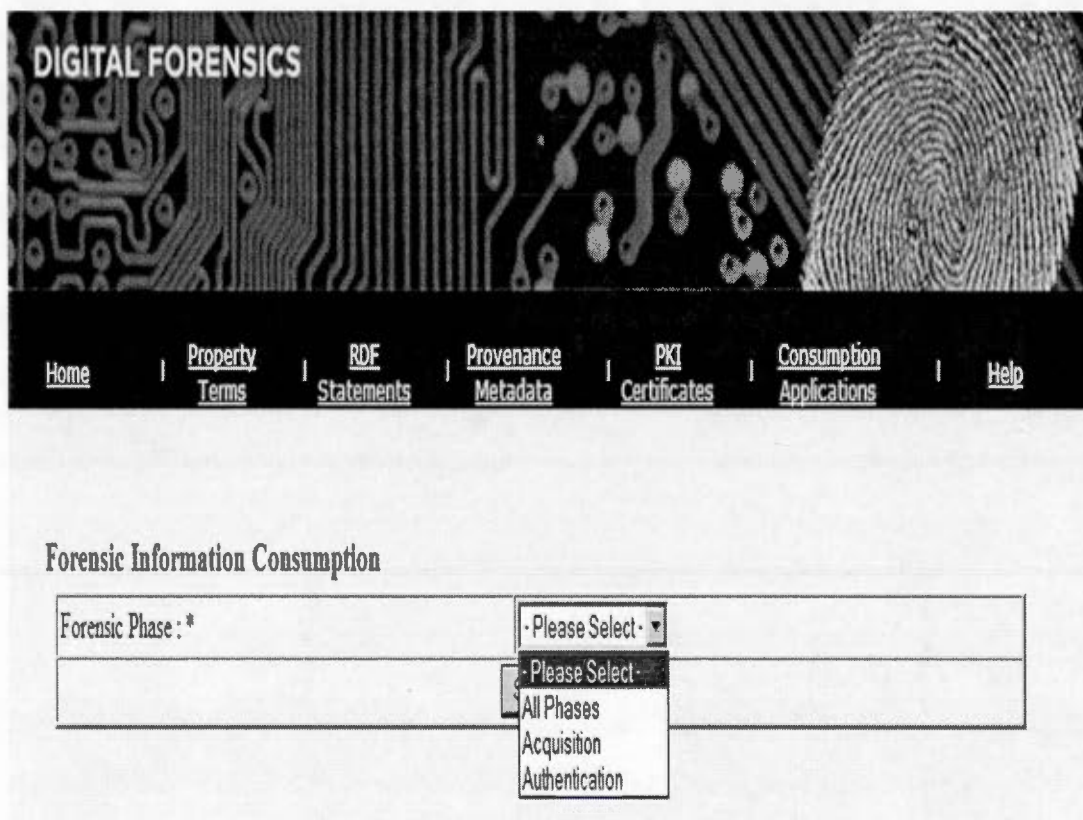
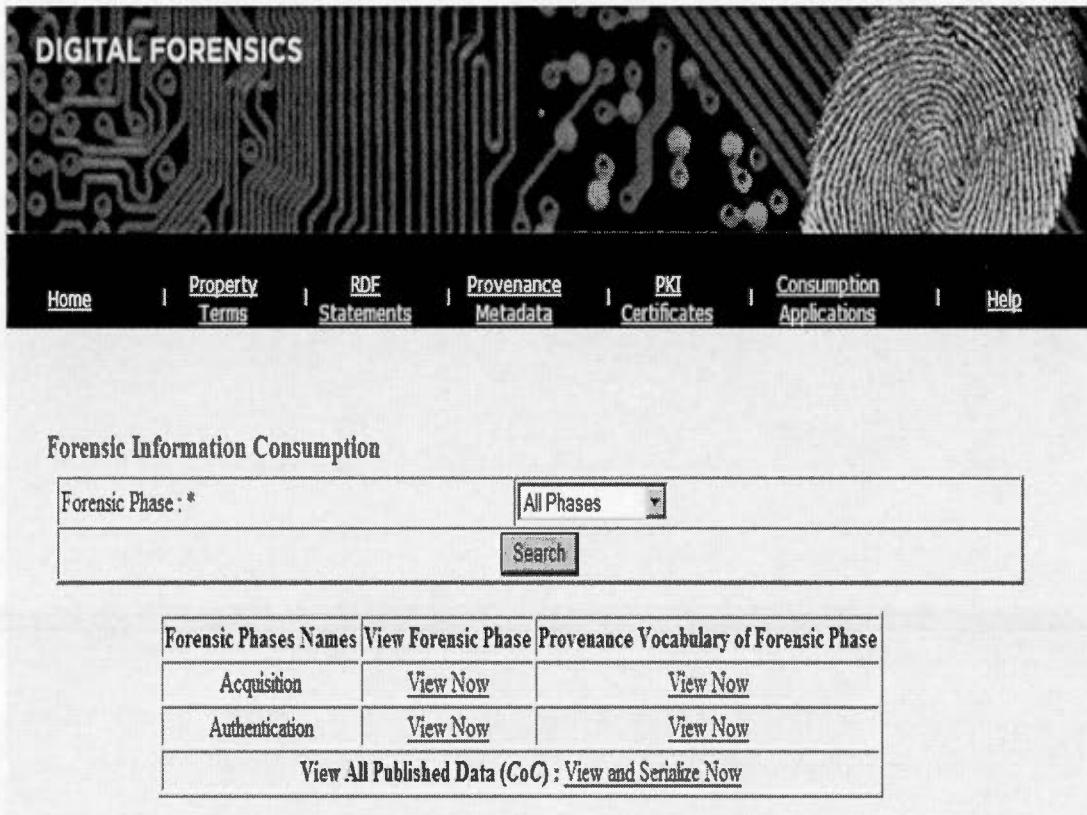


Figure 5.3 Screen for browsing

In Figure 5.3, if the consumers select “All phases”, the system will display all *e*-CoCs published by role players (technicians) during the forensic investigation. As shown in the above figure, there are currently two forensic phases published and recorded/stored by the role players on the system. The screen resulted from this selection allows consumers to visualize all RDF models related to each forensic phase (see Figure 5.4). These RDF models are those related to the forensic phase and their provenance metadata.



DIGITAL FORENSICS

[Home](#) |
 [Property Terms](#) |
 [RDF Statements](#) |
 [Provenance Metadata](#) |
 [PKI Certificates](#) |
 [Consumption Applications](#) |
 [Help](#)

Forensic Information Consumption

Forensic Phase : * All Phases

Forensic Phases Names	View Forensic Phase	Provenance Vocabulary of Forensic Phase
Acquisition	View Now	View Now
Authentication	View Now	View Now
View All Published Data (CoC) : View and Serialize Now		

Figure 5.4 Screen of all forensic phases

When consumers click on the “*view forensic phase*”, they will get information about the definition of this forensic phase, such as who published, when it was published, what is the name and its label (See Figure 4.3).

In addition, they may also discover more information about each graph apart by clicking on the link “*view provenance vocabulary*”. This will provide them more metadata information about the *e-CoC* associated to each phase as shown in Figure 5.5. These metadata may answer more questions related to the forensic and provenance information.

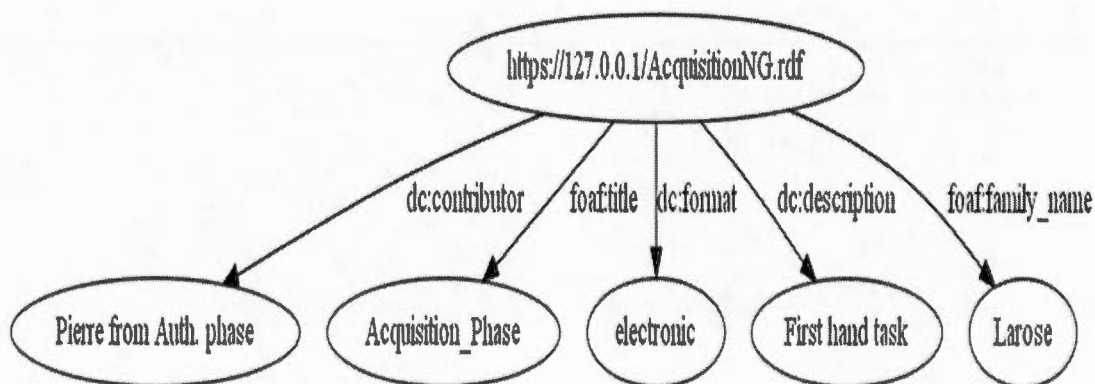


Figure 5.5 Screen for viewing provenance vocabulary

Furthermore, beside these two types of RDF models, the system can also display the data instances of *e*-CoCs related to all forensic phases stored in the system (A-Box), through the link “view and serialize”. This link can display to consumers all RDF triples of published data (*e*-CoCs) in one screen, which are added with their serialization codes. These codes serialize-down the RDF graphs (e.g., N3, Turtle, RDF/XML, etc.). The generated serialized code in the current example is the RDF/XML, because it is the most popular serialized code of the semantic web. Arguably beside JSON-Linked Data (see Figure 5.6). Next figure, Figure 5.6, shows the RDF model of published data, and Figure 5.7 shows its corresponding serialized code.

Extra information has been added to the current tangible CoC (see Chapter 4) to elaborate how the system can generate for consumers all the *e*-CoCs published by the role players. For example, new information has been published, called “*iPad*” and “*PersonalDigitalAssistant*”. The “*iPad*” has also the same serial number for the “*PDA device*”. The “*PersonalDigitalAssistant*” is another preserved media device within the preservation task. Also, besides the current forensic phase (i.e., acquisition), another forensic phase called “*authentication phase*” has been provided. The role player of this phase is called “Peter”, who authenticated the same media

device “*PersonalDigitalAssistant*”, preserved by “*Jean-Pierre*”. “*Peter*” also recorded the hash code “*0X49E9DEC3*” generated from his authentication task (i.e., assuming that the “*authenticate*” and “*hash*” terms have been defined on the T-Box level).

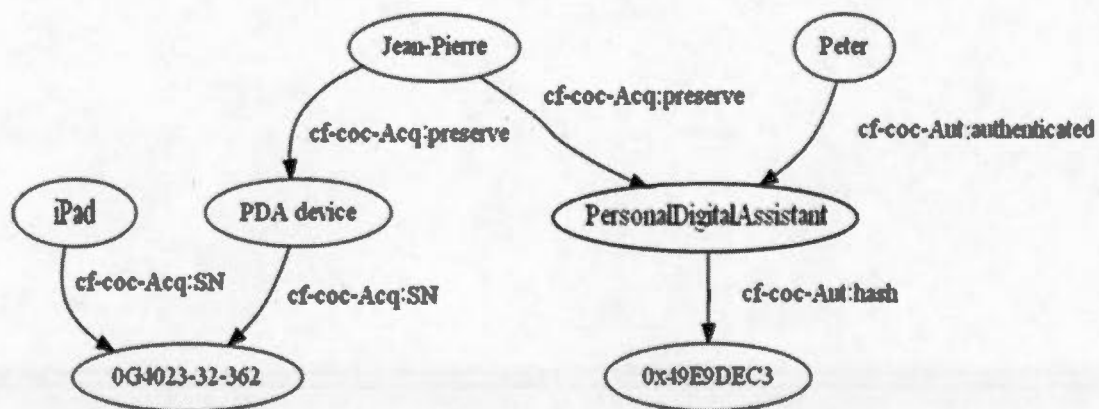


Figure 5.6 *e*-CoCs for two forensic phases

This information has been added to the graph to underline two points. Firstly, the generation of one or more *e*-CoC in one RDF model. Secondly, how two *e*-CoCs representing two different forensic phases are linked together (i.e., through “*PersonalDigitalAssistant*” object slot). This is considered as an indirect improvement that the forensic phases are dependent and their investigation tasks can be performed in collaboration between role players (i.e., a media device is firstly preserved, and then this same device is authenticated later by another role player). Authentication phase tasks will be discussed in Chapter 7.

From Figures 5.6 and 5.7, it is noticed that there are two namespaces. One for the acquisition phase called “*cf-coc-Acq:*” and contains some terms such as “*SN*” and “*preserve*”, that the role player has defined in his vocabulary to publish his own CoC information. The second namespace is for the authentication phase and called “*cf-coc-Auth:*”, which contains two terms: “*authenticated*” and “*hash*”. Using these

different views, consumers therefore can have a global virtualized picture of the *e*-CoCs published by role players (technicians) during the investigation process.

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:cf-coc-Acq="https://cyberforensics-coc/Acquisition/"
xmlns:cf-coc-Aut="https://cyberforensics-coc/Authentication/"
xmlns:rdfs="http://www.w3.org/2000/10/XMLSchema#">
  <rdf:Description rdf:about="iPad">
    <cf-coc-Acq:SN
rdf:datatype="http://www.w3.org/2000/10/XMLSchema#string">0G4023-
32-362 </cf-coc-Acq:SN>
  </rdf:Description>
  <rdf:Description rdf:about="PDA device">
    <cf-coc-Acq:SN
rdf:datatype="http://www.w3.org/2000/10/XMLSchema#string">0G4023-
32-362 </cf-coc-Acq:SN>
  </rdf:Description>
  <rdf:Description rdf:about="Jean-Pierre">
    <cf-coc-Acq:preserve rdf:resource="PDA device"/>
    <cf-coc-Acq:preserve rdf:resource="PersonalDigitalAssistant"/>
  </rdf:Description>
  <rdf:Description rdf:about="Peter">
    <cf-coc-Aut:authenticated
rdf:resource="PersonalDigitalAssistant"/>
  </rdf:Description>
  <rdf:Description rdf:about="PersonalDigitalAssistant">
    <cf-coc-Aut:hash
rdf:datatype="http://www.w3.org/2000/10/XMLSchema#string">0x49E9DE
C3 </cf-coc-Aut>
  </rdf:Description>
</rdf:RDF>
```

Figure 5.7 RDF/XML of *e*-CoC for the preservation

The above part discussed the case of the role player/judge selects “*all phases*”. Next part discusses when the role player/judge selects a specific forensic phase (e.g. ‘*Acquisition*’, ‘*Authentication*’). In the case of the second selection in Figure 5.3; if consumers select a specific forensic phase, this will help them to expand/dereference different terms described by role players. As mentioned, the root definitions of such resources are defined from a well-defined vocabulary of the semantic web. This is shown in Figure 5.8.

DIGITAL FORENSICS

Home | Property Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Forensic Information Consumption

Forensic Phase : *

Forensic Tasks For Acquisition Phase	Described by Resource(s)
Preservation	https://127.0.0.1/Vocab/Acquisition#RolePlayer https://127.0.0.1/Vocab/Acquisition#FirstResponder https://127.0.0.1/Vocab/Acquisition#DigitalMedia https://127.0.0.1/Vocab/Acquisition#SN https://127.0.0.1/Vocab/Acquisition#preservedBy https://127.0.0.1/Vocab/Acquisition#preserve

Figure 5.8 Screen showing the resources of preservation task

As shown in the above figure, the acquisition phase contains different resources defined by the role players. Let’s assume that the consumers after displaying the *e*-CoCs in Figure 5.6 need to get more information about the published resources. This screen will allow them to dereference different resources in order to get more information about the published resources.

- The “*FirstResponder*”:

When the consumers dereference this resource, the system will navigate them (follow-their-nose) to the root definition of this term.

Resource URL : https://127.0.0.1/Vocab/Acquisition#FirstResponder	
Term Type :	Class
Sub Class of :	<u>Role Player</u>

Figure 5.9 Screen for “*FirstResponder*” resource expansion

The Figure 5.9 shows that the “*FirstResponder*” term is a subclass of the “*RolePlayer*”. The latter is also an expanded resource. If consumers continue to follow-their-noses, they will get the root definition (*FirstResponder* is subclass of *RolePlayer*, which is sub-class of class *Person*) as shown in Figure 5.10 and Figure 5.11.

Resource URL : https://127.0.0.1/Vocab/Acquisition#RolePlayer	
Term Type :	Class
Sub Class of :	<u>Person</u>

Figure 5.10 Screen for “*RolePlayer*” resource expansion

Finally, by expanding “*Person*”, the consumers arrive to the root definition of the “*FirstResponder*”, as shown in Figure 5.11.

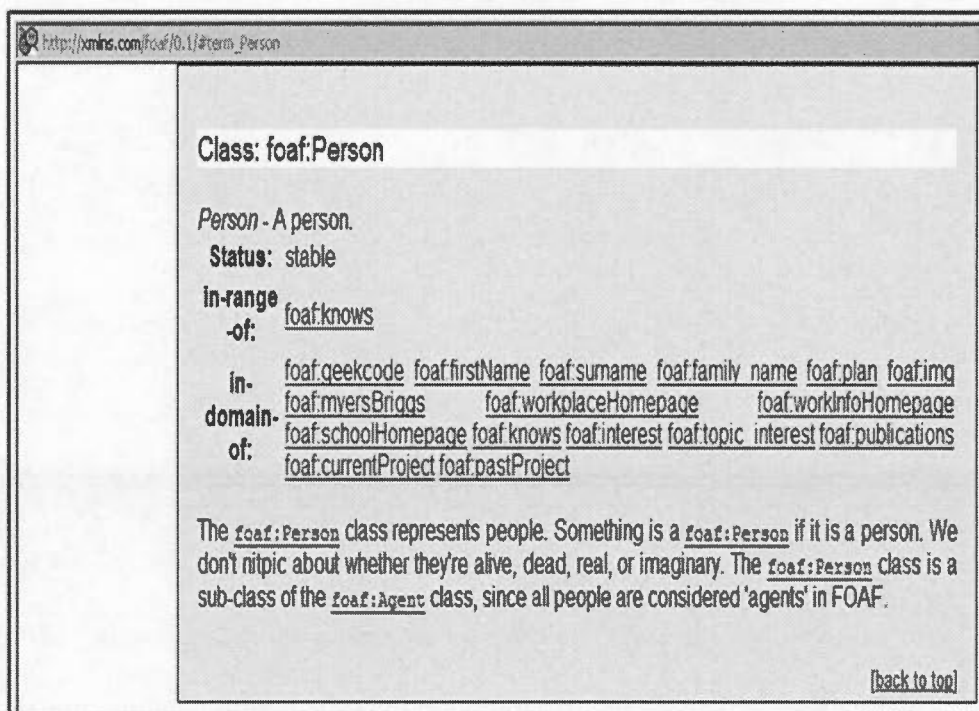


Figure 5.11 Screen for “*Person*” definition

- The “*RolePlayer*” :
It is the same idea as that of “*FirstResponder*” term. See Figure 5.10 and 5.11.
- The “*DigitalMedia*” :
When consumers dereference this resource, the system will navigate them to the root definition of this term. As has been explained, it is the same idea of expansion till reaching the root definition of the resource.

All the above terms are of type class. The next part of the dissertation discusses resources of type property, and how they are presented to the consumers. In this part, more information is provided to consumers, such as domain and range of the term:

- The “SN” :

The “SN” term is a property defined by the role player of the acquisition phase. This predicate is the serial number of a digital device and it is tagged with “*owl:InverseFunctionalProperty*” constructor (see Figure 5.12). The reason for using this constructor with this predicate will be explained in Section 5.4.

Resource URL : <https://127.0.0.1/Vocab/Acquisition#SN>

Term Type :

Property , Inverse Functional Property

Sub Property of :

identifier

Domain :

DigitalMedia

Range :

Literal

Below are the instances of class DigitalMedia	Predicate	Below are the instances of class Literal
PDA device (Acquisition Phase)	cf-coc-Acq-SN	0G4023-32-362 (Acquisition Phase)
iPad (Acquisition Phase)	cf-coc-Acq-SN	0G4023-32-362 (Acquisition Phase)

Figure 5.12 Screen for “SN” resource expansion

As shown in Figure 5.12, the domain and range of the property “SN” are mentioned, and can also be expanded to their root definitions. Also, this screen shows all the related instances of this property and for which forensic phase such instances have been defined. In addition, the RDF graph of those

instances is also drawn on the same screen, as shown in the figure below (Figure 5.13).

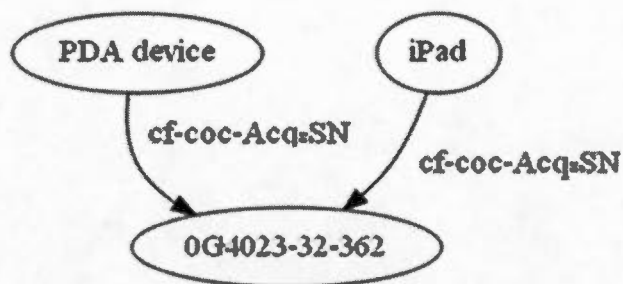


Figure 5.13 Screen for RDF instances of “SN” property

- The “*preserve*” term :

The “*preserve*” term is a property defined by the role player of the acquisition phase. This chapter tags this term with “*owl:FunctionalProperty*” constructor (see Figure 5.14) to illustrate the case if the role player selects this constructor instead of “*owl:InverseFunctionalProperty*” as explained in Chapter 4. This will be explained in Section 5.4.

Also, this screen shows all the related instances of this property and for which forensic phase such instances have been defined. In addition, the RDF graph of those instances is also drawn on the same screen, as shown in the figure below (Figure 5.14).

Resource URL : https://127.0.0.1/Vocab/Acquisition#preserve		
Term Type : Property , Functional Property		
Sub Property of : <u>made</u>		
Domain : <u>FirstResponder</u>		
Range : <u>DigitalMedia</u>		
Inverse of : <u>preservedBy</u>		
Below are the instances of class FirstResponder	Predicate	Below are the instances of class DigitalMedia
Jean-Pierre (Acquisition Phase)	cf-coc-Acq-preserve	PDA device (Acquisition Phase)
Jean-Pierre (Acquisition Phase)	cf-coc-Acq-preserve	PersonalDigitalAssistant (Acquisition Phase)

Figure 5.14 Screen for “*preserve*” resource expansion

As well, the instances of the acquisition forensic phase are also shown in Figure 5.14, accompanied by its corresponding RDF model (see Figure 5.15).

In Figure 5.15, the role player “*Jean-Pierre*” is described as a resource in the definition of the acquisition phase. The system recognizes the ontologies and the proprietary terms published by each role player. PDA device and PersonalDigitalAssistant are instances of the class “*DigitalMedia*”.

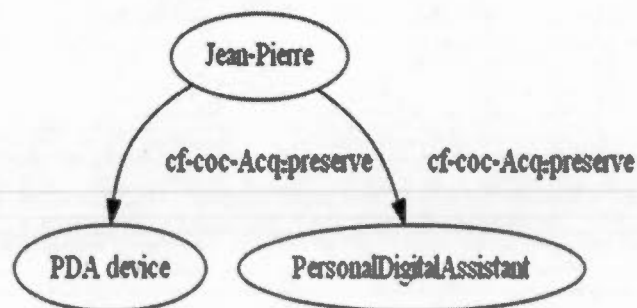


Figure 5.15 Screen for RDF instances of “*SN*” property

The system detects this information by reading the fields of the digital certificate submitted by the role player (see Chapter 6). One of these fields is the name of its owner. Thus, the system reads the certificate to identify who worked on different ontologies.

On the other hand, the “*PersonalDigitalAssistant*” is another resource published during the acquisition phase, not the authentication one. This is due to the fact that the role player of the acquisition phase is the one who starts to use this resource to describe this media device. Later, “*Pierre*” who is the role player of the authentication phase, can use the same resource published in the acquisition phase to perform another task called the authentication.

5.3 Crawling

Browsing is not the only pattern to discover or request information. Crawling or searching by a keyword can be another pattern to discover such information. Crawling is searching by a specific keyword (see Section 2.4.2). It allows consumers to search specific information. The information requested will be displayed within its RDF triple and can be one of the three slots of the RDF (see Figure 5.16).

DIGITAL FORENSICS

Home | Property Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Enter a Resource/Literal Keyword for Crawling Triples: *

(Enter % to Crawl all Triples)

Crawl Now

Figure 5.16 Screen for crawling resources/literals

The system also allows consumers to display all triples stored in the system through the delimiter symbol “%”. Figure 5.17 shows a screen resulted from crawling triples using the keyword:

DIGITAL FORENSICS

Home | Property Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Enter a Resource/Literal Keyword for Crawling Triples: *

(Enter % to Crawl all Triples)

Crawl Now

Triples		
Jean-Pierre (Acquisition)	preserve (Acquisition)	PDA device (Acquisition)
Jean-Pierre (Acquisition)	preserve (Acquisition)	PersonalDigitalAssistant (Acquisition)

Figure 5.17 Screen for crawling the “*preserve*” term

In this figure, a consumer can crawl using the “*preserve*” property. The system then displays all role players who preserved different digital media. Similarly, when a consumer crawls using “*Jean-Pierre*”; he will get the same results appearing above (i.e., what are the different digital devices that were considered by “*Jean-Pierre*”).

Using crawling, the requested information may appear in one or more slot at the time. This means that all related triples containing this keyword will also be displayed (See Figure 5.18).

The screenshot shows a web application titled "DIGITAL FORENSICS" with a header containing navigation links: Home, Property Terms, RDE Statements, Provenance Metadata, PKI Certificates, Consumption Applications, and Help. Below the header, there is a search section with the prompt "Enter a Literal Keyword for Crawling Triples: *" and a sub-prompt "(Enter % to Crawl all Triples)". A text input field contains the keyword "PDA device", and a "Crawl Now" button is positioned below it. The results are displayed in a table titled "Triples".

Triples		
PDA device (Acquisition)	SN (Acquisition)	0G4023-32-362 (Acquisition)
Jean-Pierre (Acquisition)	preserve (Acquisition)	PDA device (Acquisition)

Figure 5.18 Screen for crawling using the “*PDA device*”

As shown in Figure 5.18, the consumer crawls by using the “*PDA device*” keyword. The system displayed two different triples. First triple shows the “*PDA device*” appeared on the subject slot and the second triple the “*PDA device*” appeared on the object slot. In addition, each slot is also appended by the name of the forensic phase to which the crawled term belongs.

5.4 Reasoning

Another pattern provided in the CF-CoC system, is the possibility for consumers to infer implicit information from the published triples. Inference rules are implemented within this section according to the rules mentioned in section 2.2.1.4.

As shown in Figure 5.19, the system asks the consumer on which ontology he wants to run the reasoning engine.

The screenshot shows a web application titled "DIGITAL FORENSICS" with a background image of a circuit board and a fingerprint. A navigation bar contains links: Home, Property Terms, RDF Statements, Provenance Metadata, PKI Certificates, Consumption Applications, and Help. Below the navigation bar, the section "Forensic Information Consumption" is displayed. It features a form with a label "Reason Over Forensic Phase : *" and a dropdown menu. The dropdown menu is open, showing options: "Please Select", "Acquisition", and "Authentication". A "Reasoning Now" button is located below the form.

Forensic Information Consumption	
Reason Over Forensic Phase : *	- Please Select -
	- Please Select -
	Acquisition
	Authentication
Reasoning Now	

Figure 5.19 Screen for reasoning on a forensic phase

Consumer selects from this screen the forensic phase on which he wants to infer extra information from the published triples. Based on the same example provided in Chapter 4, if the consumer selects the acquisition phase, the reasoning engine will run on all defined terms and their related instances in this phase.

The constructors that are not yet explained, are the two constructors “*owl:FunctionalProperty*” and “*owl:InverseFunctionalProperty*” associated to “*preserve*” and “*SN*”, respectively.

- The “*preserve*” term :

The “*preserve*” property is a predicate defined using “*owl:ObjectProperty*” and at the same time it is also defined using the constructor of “*owl:FunctionalProperty*”. When a property is tagged using this latter constructor, it means that for each subject in the triple where the “*preserve*” property is a predicate there can be at most one object. (Note: this is an example of using the *owl:FunctionalProperty* with the “*preserve*” term. In chapter 4 the *owl:InverseFunctionalProperty* is used with the “*preserve*” term, but in this illustration, the *owl:FunctionalProperty* is used. Also, see Table 2.2 and Table 2.3).

Referring to the Figure 5.15, it is shown that the role player “*Jean-Pierre*” preserved two media devices, “*PDA device*” and the “*PersonalDigitalAssistant*”. Both triples are described using the “*preserve*” property, which is tagged the “*owl:FunctionalProperty*” constructor. Thus, the system will consider and display to the consumers that both terms have the same identity, but they are presented using two different syntaxes.

To illustrate on this idea using the entailment rule of Table 2.3:

If $p(x,y)$ and $p(x,z) \Rightarrow y=z$

If $\text{preserve}(\text{Jean-Pierre}, \text{PDA device})$ and $\text{preserve}(\text{Jean-Pierre}, \text{PersonalDigitalAssistant}) \Rightarrow \text{PDA device} = \text{PersonalDigitalAssistant}$

This rule depends totally on how the role players defined the “*preserve*” term. As mentioned earlier in this dissertation, it is assumed that the role players are always aware of the lightweight constructors of RDFS++. In this case, the role players restricts, that each role player can use the “*preserve*” term with only

one device. According to this example, when a role player uses another term syntax to describe the same device then both strings will be equated to each others.

Because the term can be used by one or more role players from other forensic phase, the role players may use the constructor *owl:InverseFunctionalProperty* to restrict the preservation task to only the owner of the term. When a property is tagged using this latter constructor, it means that for each object in the triple where the “*preserve*” property is a predicate there can be at most one subject. Thus, when consumers consume triples containing this predicate, the system will notify them that all related subjects to this term are the same.

- SN:

The “*SN*” property is a predicate defined using “*owl:ObjectProperty*” and at the same time it is also defined using the constructor of “*owl:InverseFunctionalProperty*”. When a property is tagged using this latter constructor, it means that in the triple where the “*SN*” property is a predicate, the object will have one and only one subject (See Table 2.2 and Table 2.3).

Referring to the Figure 5.13, the literal object “*OG4023-32-362*” is the same for two different media devices, the “*IPad*” and the “*PDA device*”. Both triples are described using the “*SN*” property, which is tagged by the “*owl:InverseFunctionalProperty*” constructor. Thus, the system will consider and display to the consumer that both terms are semantically the same, but they are presented using two different syntaxes.

To illustrate on this idea using the entailment rule of Table 2.3:

$$\text{If } p(y,x) \text{ and } p(z,x) \Rightarrow y=z$$

If $SN(IPad, 0G4023-32-362)$ and $SN(PDA\ device, 0G4023-32-362) \Rightarrow$
 $iPad = PDA\ device$

Because the range of the object slot, where “*SN*” is a predicate, is of type *Literal*, and is used to identify one and only one device, the role players may use to tag this property using the “*owl:InverseFunctionalProperty*” (i.e., to restrict that all literals provided by “*SN*” predicate uniquely identify one subject). Figure 5.20 shows a screen displaying the result from both constructors.

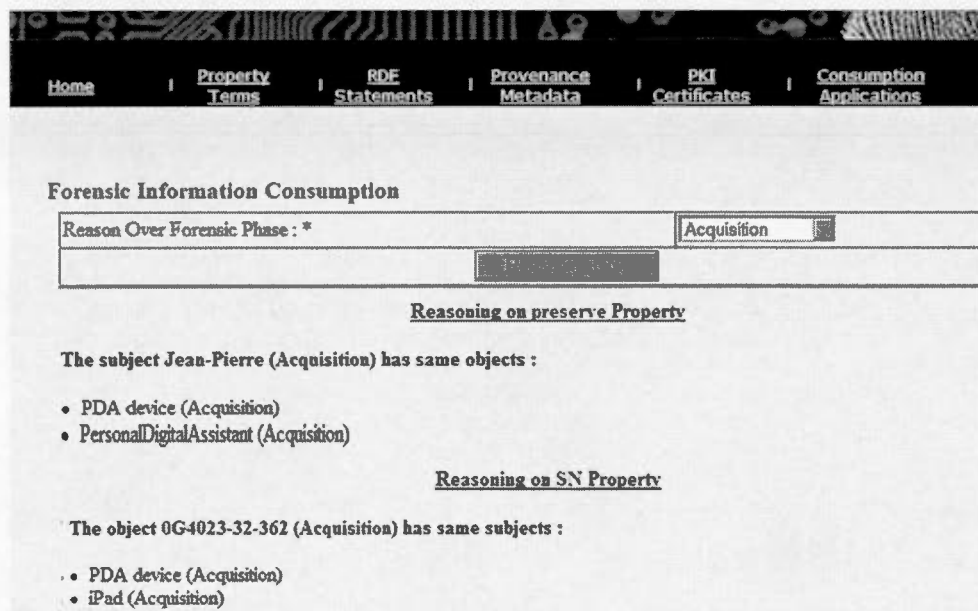


Figure 5.20 Screen for reasoning on “*preserve*” and “*SN*”

Thus, from both terms, the system can deduce implicit information, that the three mentioned terms: “*IPad*”, “*PDA device*”, and “*PersonalDigitalAssistant*” are all the same. The reason is that “*PDA device*” is “*PersonalDigitalAssistant*”, and “*IPad*” is “*PDA*” device, implies that “*IPad*” is “*PersonalDigitalAssistant*”. Based on the role player definitions of these predicates, the three terms are describing the same world

concept, but using different terms. Thus, *owl:sameAs* can be used to map these terms together.

Other information can also be inferred by the system:

- “*Jean-Pierre*” is of type “*FirstResponder*” and “*FirstResponder*” is a subclass of type “*RolePlayer*”, implies that “*Jean-Pierre*” is also of type “*RolePlayer*”.
- “*Jean-Pierre*” preserve “*IPad*” and “*preserve*” is a sub-property of “*made*” property, implies that “*Jean-Pierre*” made “*IPad*”.
- “*preserve*” domain is “*FirstResponder*” and “*Jean-Pierre*” “*preserve*” “*PersonalDigitalAssistant*”, implies “*Jean-Pierre*” is a “*FirstResponder*”.
- “*preserve*” range is “*DigitalMedia*” and “*Jean-Pierre*” “*preserve*” “*PersonalDigitalAssistant*”, implies “*PersonalDigitalAssistant*” is a “*DigitalMedia*”.

These are not the only information that the system can infer. More triples could be inferred, depending on the number of constructors defined and the number of resources and literals published by the role players.

5.5 Querying

The last pattern that will be discussed in this chapter is querying. Usually, the word query in the LD refers to SPARQL query to retrieve explicit information from the RDF store. The remaining part will discuss how a consumer can retrieve explicit information in the CF-CoC without writing down SPARQL code.

The idea is based on searching a specific literal (in object slot) or resources (in subject, predicate, or object) on each slot. This will be very useful when the consumer

do not know what literal/resource he should type to crawl certain information. This may happen at the beginning of his survey over the cyber criminal case.

Thus, a consumer can go to this screen to display explicit information published by the role players. Each slot they select can display all information related to this slot. He can also display the default option “All” to display all triples in the system (see Figure 5.21)

Subject	Predicate	Object
- All Subjects (default) -	- All Predicate (default) -	- All Objects (default) -
Query Now		
PDA device	SN	0G4023-32-362
iPad	SN	0G4023-32-362
Jean-Pierre	preserve	PDA device
Jean-Pierre	preserve	PersonalDigitalAssistant

Figure 5.21 Screen showing all RDF triples

If for example, a consumer needs to query using the object slot, he can select a specific object from this slot, and the system will display all related triples, where this value of query appears as an object in all triples (See Figure 5.22).

All combinations are possible to query different information. The only difference between this pattern and the crawl pattern is in the awareness of consumers about the information published in the system. If the consumer knows what he wants to find, he can then use the crawl consumption pattern, and if he does not know about what are the resources published by the role players, he can start to query from a set of mentioned resources and literals.

DIGITAL FORENSICS

Home | Property Terms | RDF Statements | Provenance Metadata | PKI Certificates | Consumption Applications | Help

Forensic Information Consumption

Subject	Predicate	Object
Jean-Pierre	- All Predicate (default) -	- All Objects (default) -
<input type="button" value="Query Now"/>		
Jean-Pierre	preserve	PDA device
Jean-Pierre	preserve	PersonalDigitalAssistant

Figure 5.22 Screen for querying upon subject slot

5.6 Conclusion

This chapter discussed different consumption patterns provided by the CF-CoC system. The aim of this chapter was to explain how the system can help each role player to consume the published resources published by other role players and aid consumers in the court to understand and discover all published information.

The consumption patterns discussed in this chapter are: browsing, crawling, reasoning and querying. Browsing is to navigate through different resources and expand them to understand their root definitions. Crawling is to search by a specific keyword. Reasoning is to infer implicit information from explicit information. Finally, querying is to search by keywords already displayed by the system.

The next chapter will discuss how the published linked data are bent to be consumed on a closed scale using PKI.

CHAPTER VI

LINKED CLOSED DATA USING PUBLIC-KEY INFRASTRUCTURE

6.1 Introduction

This chapter explains how the Linked Data Principles (LDP) that are used to publish and represent information on a large scale (i.e., the case of LOD project) can be bended to publish such information on a closed scale (i.e., this scale is called Linked Closed Data, LCD).

As mentioned, the position of this chapter in this dissertation does not mean that this task comes after the representation task and consumption of resources. Indeed, the task discussed in this chapter comes before starting the investigation process and it is realized all over the forensic investigation, since the seizure of events, until the consumption of published resources by judge in a court of law. The restriction and accommodation of resources on closed scale will be accomplished by using the digital certificates.

The digital certificates will be generated using the OpenSSL tool²⁷. Three types of certificates will be generated. The Certificate Authority (CA) certificate, called also the self-signed certificate or root certificate, the server certificate, and the client certificate.

²⁷ <https://www.openssl.org/>

Usually, the CA certificate can be generated for public or private community in order to sign server and client certificate. The CA certificate generated for each community is essentially the same, except that it differs and based on the following fact: “On whom the requesters put their trusts”. Therefore, the generation of CA certificates and the signing of client and server certificates, follow the same procedures with all communities.

In the current context, a scenario (see Section 3.5) has been proposed among a neutral side that hosts the CF-CoC, role players, and judge to share digital certificates. This scenario is enrolled between them under the assumption that the neutral part is responsible to select the CA provider. After a neutral part selects the CA provider, it requests a server certificate for its CF-CoC system and requests a client certificate for the judge. Role players are also responsible to send their requests for client certificates to the same CA provider. Role players should have client certificates to access and consume such information.

Due to the cost of issuing digital certificates from a public reputed institution, this dissertation assumes a virtual/imaginary institution called CF-CA. Its CA certificate will be generated manually to sign both server and client certificates.

Generally, the CF-CoC system resides on a unique domain/namespace on the web cloud owned by a neutral side. The latter installs a server certificate to secure the published information. The CA is responsible to issue and signs the server certificate for the neutral side, a client certificate for judge, and client certificates for all role players.

Before the investigation process takes place, the neutral side receives a list of all role players who are going to participate in the forensic investigation. The neutral side in turn sends this list to the CA. The latter compares this list with the requests received from the role players and issues their client certificates.

6.2 Creation phases

Generally, the creation of a digital certificate passes by four phases (see Figure 6.1) using the OpenSSL tool. Firstly, the certificate requester generates its own pair of keys (i.e., *.key* file), then creates a request (i.e., *.req* or *.csr* format file) to the trusted party to issue for him a certification file (i.e., *.crt*). Also, same procedures can be applied to the CA. The only difference for generating the CA certificate will be on its certification request. The request is sent and signed by the CA itself (i.e., that is why this type of certificate is called self-signed certificate).

The trusted party (i.e., CA) signs the requests and issues the corresponding certificates using its own private-key. The created certificate is then transformed to an exportable format (i.e., *.p12* format) for sending it to the requesters (i.e., neutral side that hosts the system and role players).

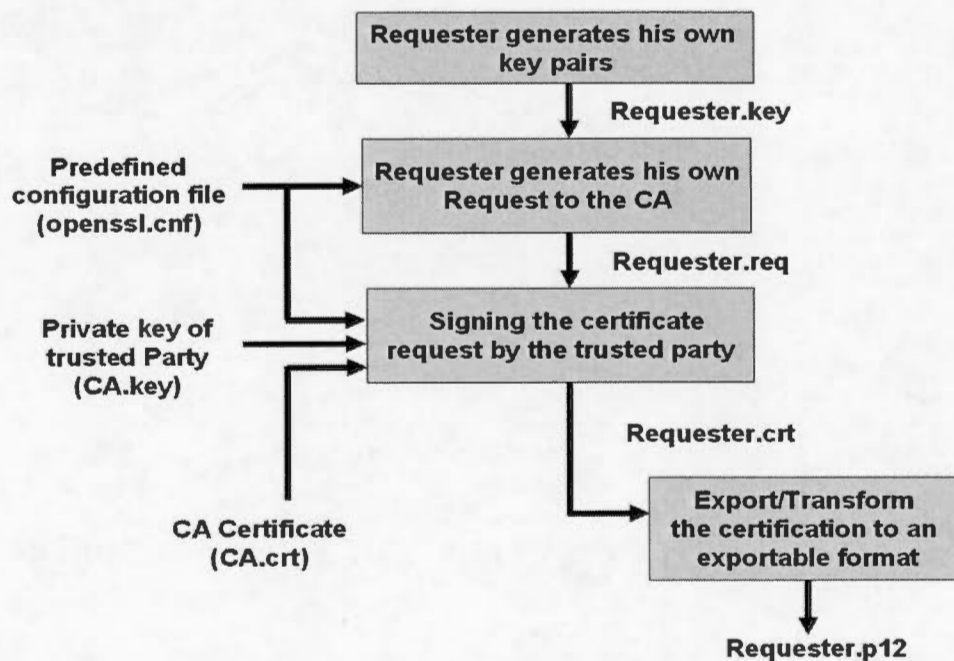


Figure 6.1 Procedures for creating a digital certificate using the OpenSSL tool

6.2.1 Creation of self-signed certificate

Before starting, the CA key is generated; *RootCA.key* will be of length 2048 bits (256 bytes):

```
openssl genrsa -out RootCA.key 2048
```

The *RootCA.key* is then used to generate the certificate request *RootCA.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=CA Provider), and the common name of the certificate (i.e., CN=CF-CA) (see Figure 6.2).

```
openssl req -new -key RootCA.key -out RootCA.csr -config openssl.cnf -subj  
"/C=CA/O=CA Provider Institution/CN=CF-CA/"
```

After generating the *RootCA.csr*, the request is signed using the *RootCA.key* to generate the requested certificate (*crt* format, *RootCA.crt*). In this type of certificate, the CA itself will sign the certificate, that's why it is called a self-signed certificate:

```
openssl req -x509 -days 365 -in RootCA.csr -out RootCA.crt -key RootCA.key -config  
opensslCA.cnf -extensions v3_ca
```

Finally, the exportable format *.p12* is generated to transform the *RootCA.crt* into an exportable format *RootCA.p12*

```
openssl pkcs12 -export -in RootCA.crt -inkey RootCA.key -certfile RootCA.crt -out  
RootCA.p12
```



Figure 6.2 The CA self-signed certificate

6.2.2 Creation of server certificate

The server certificate is created for two goals: it lets the role player ensure the identity of the server, and it is used to check for the client certificate (see Figure 6.3).

We assume that the server name corresponds to a domain²⁸. This certificate will be issued for the neutral side (that hosts the system) to install it on his server. This server will host the CF-CoC system, which will be used by the role player. Thus, the CA will issue and sign a certificate for this domain name.

²⁸ Domain owned by Tamer Gayed : www.cyberforensics-coc.com

Firstly, the *Server.key* is generated using the following command:

```
openssl genrsa -out Server.key 2048
```

The *Server.key* is then used to generate the certificate request *Server.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=CA Provider), and the common name of the certificate (i.e., CN=Neutralside).

```
openssl req -new -key Server.key -out Server.csr -config openssl.cnf -subj  
"/C=CA/O=CA Provider/CN=Neutralside/"
```

After generating the *Server.csr*, the request is signed using the CA certificate *RootCA.crt* and the key *RootCA.key* to generate the requested certificate (i.e., *Server.crt*).

```
openssl ca -days 365 -in server.csr -cert RootCA.crt -out Server.crt -keyfile  
RootCA.key -config opensslserver.cnf -extensions server
```

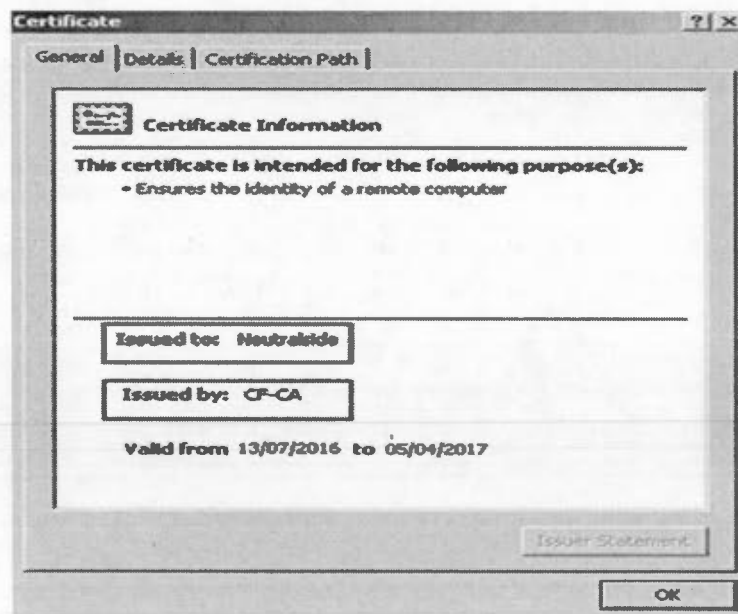


Figure 6.3 The server digital certificate

Because the server certificate is signed by the CA, the *openssl* command uses a build-in parameter called 'ca' to declare that the server certificate will be signed by the CA using its key (i.e., *RootCA.key*).

6.2.3 Creation of client certificate

The role player authenticates himself to the server through the client certificate. Role player can be a technician, prosecutor or defender. Without this certificate, the role player will not be able to access the CF-CoC system to construct ontologies for each forensic phase and publish resources (see Figure 6.4).

Firstly, the *Client.key* is generated using the following command:

```
openssl genrsa -out Client.key 2048
```

The *Client.key* is then used to generate the certificate request *Client.csr* by providing the country name (i.e., C=CA), the organization name (i.e., O=CA Provider), and the common name of the certificate (i.e., CN=Jean-Pierre).

```
openssl req -new -key Client.key -out Client.csr -config openssl.cnf -subj  
"/C=CA/O=CA Provider /CN=Jean-Pierre/"
```

After generating the *Client.csr*, the request is signed using the CA certificate (*RootCA.crt*) and key (*RootCA.key*) to generate the requested certificate (i.e., *Server.crt*).

```
openssl ca -days 365 -in Client.csr -cert RootCA.crt -out client.crt -keyfile RootCA.key  
-config opensslclient.cnf -extensions client
```

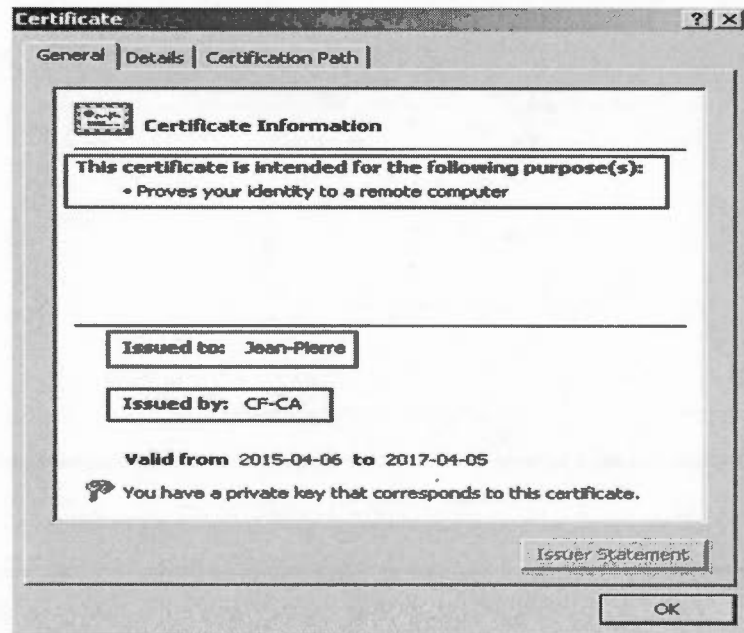


Figure 6.4 The client digital certificate

As shown in Figures 6.2, 6.3, and 6.4, each certificate has its own purpose. The purpose of a certificate depends on its type. The certificate type is defined using the *-extension* during the creation of *.crt* certificate. The *-extension* parameter calls the proper module for each certificate type. For example, it calls the *opensslCA.cnf*, *opensslServer.cnf*, and *opensslClient.cnf* for the CA, server, and client certificates, respectively. However, the *openssl.cnf* contains the general configuration of all types of certificates.

6.3 Installation of the digital certificates

Before installing the certificates, the CA sends to the neutral side and role players their own certificates. The neutral side installs his certificate on his server and role players install their certificates on their browsers.

6.3.1 Installation of self-signed certificate:

Because the CA certificate in this context is created manually and not issued by a public reputed institution, it will not be recognized by the client browsers. Therefore, the client and server certificates will not work since they are signed by custom CA provider. Thus, the CF-CA sends its self-signed certificate (i.e., *.p12* format without the private-key of the CA certificate) to the neutral side (i.e., server) and role players (i.e., clients) to install it on their machines.

By clicking on the *.p12* file (i.e., exportable format), a wizard will be launched to install the CA certificate under the trusted root folder of the current browsers for both server and clients. By firstly installing the CA self-signed certificate (i.e., CF-CA), the browsers of clients and server machines will automatically identify the issuer (i.e., custom provider) of the client and server certificates.

6.3.2 Installation of server certificate

The CA sends the server certificate to the judge. The latter then starts the installation of the server certificate. The installation of the server certificate on windows operating system passes by two phases:

- Running the Microsoft Management Console (MMC)²⁹.
- Installing the root certification authority certificate manually³⁰.
- Installing the server certificate³¹.

²⁹ <https://msdn.microsoft.com/en-us/library/ms751408.aspx>

³⁰ <https://support.microsoft.com/en-us/kb/186812#/en-us/kb/186812>

³¹ <https://support.microsoft.com/en-us/kb/892987>

Installation of this type of certificate is not accomplished by a judge. It is performed only once by a neutral side. After installing the server certificate, the neutral side is viewed only over a secure channel because it is secured using Secure Socket Module (SSL). Next figure (Figure 6.5) shows what happens when the host name is accessed without secured HTTP (i.e., `http://Neutralside`).

As shown in next figure, the server is not accessible through the classic HTTP. Page should be accessed through secure channel (i.e., SSL) using HTTPS (`https://Neutralside`).

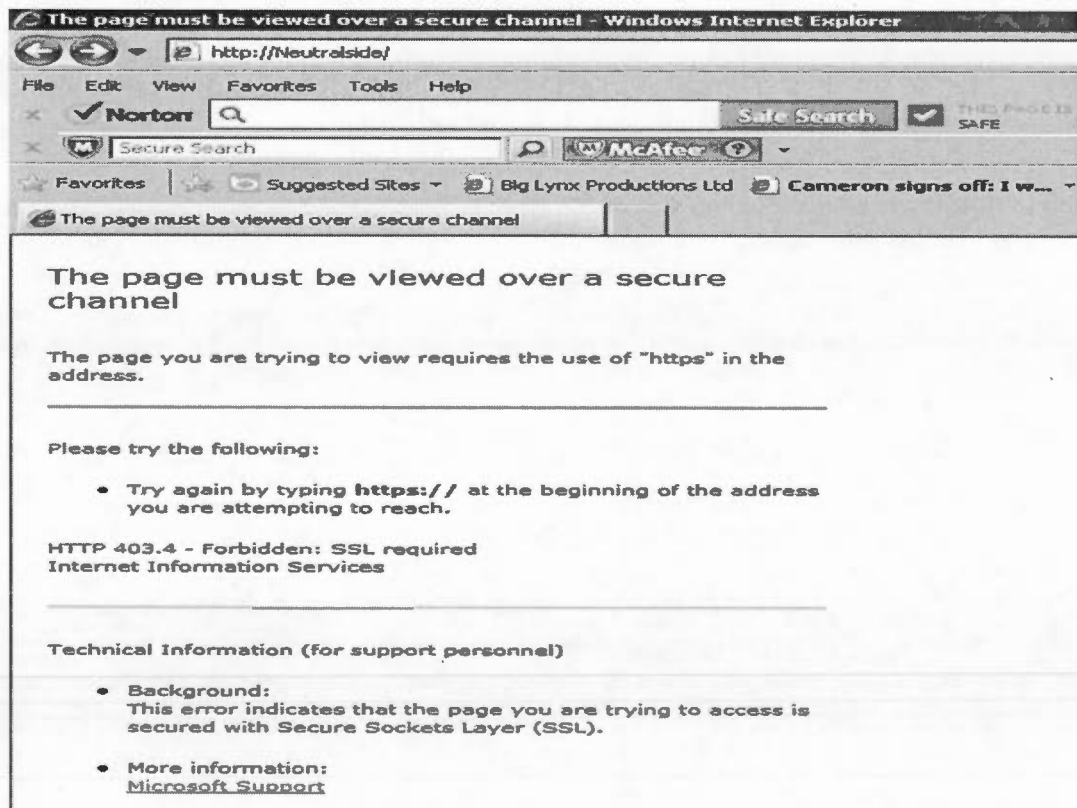


Figure 6.5 Access to CF-CoC host server using HTTP

6.3.3 Installation of client certificate

Once the page is accessed through a secure channel, it requires client digital certificate (i.e., role player or judge). If the client certificate is not installed on the machine that is trying to access this secure page, the local browser of the machine will ask the user (i.e., role player or judge) to install/have a client certificate. See Figure 6.6.

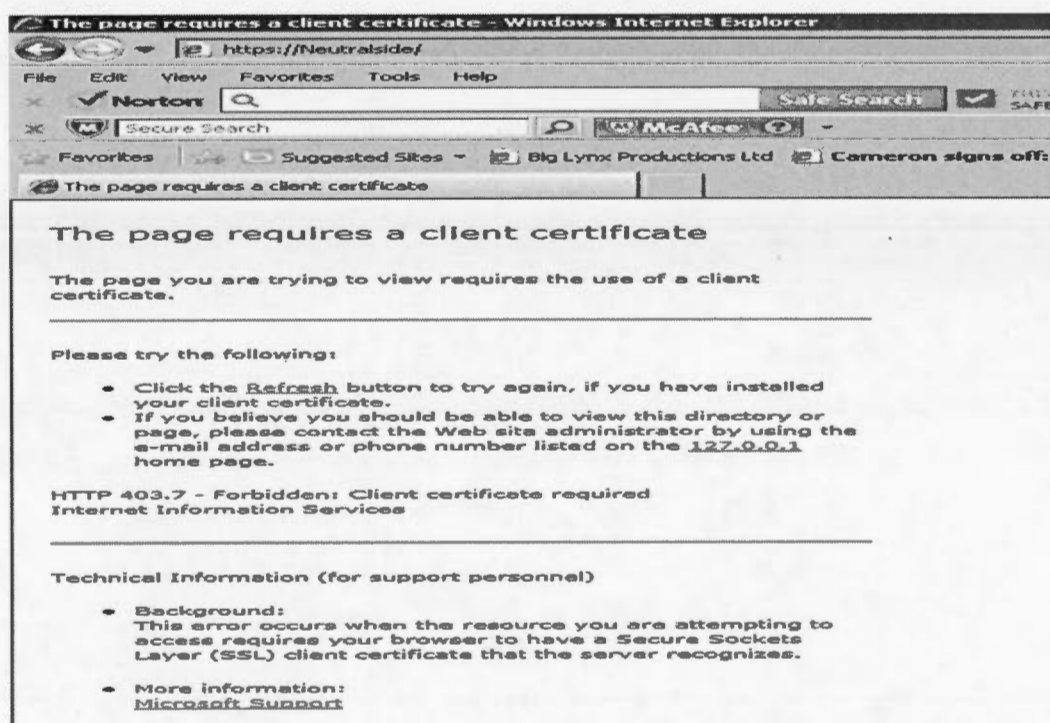


Figure 6.6 Access to CF-CoC host server using HTTPS

The installation of the client certificate is similar to the installation of server certificate, except that in this case the wizard installs the certificate in the client/personal folder of the browser.

6.4 Working scenarios of the digital certificates

This section explains two scenarios, each of them following the way of digital certificates are working from certain point of view. First scenario illustrates in technical details of the abstract scenario mentioned in Chapter 3, Section 3.5:

1. Assuming that the side that hosts the CF-CoC already selected the issuer institution and made a request for a server certificate for its host.
2. Technicians, who are authorized to gather information, send requests to the neutral side that hosts the CF-CoC system, to issue client certificates. In turn, this neutral side sends such requests to the CA, in order to issue for the technicians the clients certificates.
3. Meanwhile, each role player generates a public-private-key pair ($\{K_{U-P}, K_{R-P}\}$), where P is all information identifying the player (i.e., R is private, and U is public).
4. Each role player stores the private-key in a secure storage to keep its integrity and confidentiality, and then sends a request containing the public-key K_{U-P} and P to the CA selected by the neutral side.
5. The player's public-key and its identifying information P are signed by the CA authority using its ($\{K_{R-CA}\}$) private-key.
6. The resulting data structure is returned to the role player. $R-CA \{P, K_{U-P}\}$ is called the public-key certificate of the role player, and the authority is called a public-key certification authority (i.e., symbols outside brackets mean the signature of the data structure). The latter is installed together with the client certificate.
7. After technicians finish their tasks and publish the information (e-CoC), they communicate with the prosecutor. The latter sends a request to the neutral side for a client certificate to start using CF-CoC system (same steps from 2-6).

8. After the prosecutor receives the certificate, he starts to use the CF-CoC system to consume the published information. Prosecutor has a duty to build and publish the proofs (inculpatory evidence) against the accused.
9. After the prosecutor finishes his task, he communicates with the defense, which also sends a request to the neutral side for a client certificate to start consuming the published information.
10. If the defense disputes the evidence by showing counter evidence to thwart prosecutor (exculpatory evidence), the judge is engaged and sends a request to the neutral party for a client certificate to start using CF-CoC system (same step from 2-6).
11. After Judge receives the certificate, he starts to use the CF-CoC to consume the information published by the technicians, prosecutor and defenders and call the technicians to listen their testimonies about the e-CoC.

The second scenario explains how role players and judge access the CF-CoC system using their client certificates:

1. A role player/judge accesses the site by typing the URL of the server using the secure socket channel (see Figure 6.5).
2. Because the remote server (i.e., where the CF-CoC web application is hosted) owns a server certificate, it requires then that each client wishing to access this domain, also owns a client certificate owned by the same trusted party (In this case, the CF-CA), otherwise the browser responds with a blank page asking to install client certificates (see Figure 6.6).
3. Once the server identifies the client certificate, it redirects the client to the CF-CoC web application (see Figure 6.7).
4. The role player, at this time, accesses the application. He starts publishing/consuming the ontologies and creating terms concerned with the forensic phase in hand (see Chapter 4).

As it is shown in next, Figure 6.7, the server certificate is installed and shown at the top of the screen as a yellow lock. By clicking on the lock, it will show who issued the certificate (i.e., CA) for this page and to whom it was issued.

Once the role player finishes the publication task, the resources is available to the judge for consumption, as he owns a server certificate of the server, which allows him to view and access such resources published on his server. For example, the terms defined in Chapter 4, such as “*preserve*”, “*DigitalMedia*”, etc., will be resolvable to more extra resources on the same domain (e.g., “*preservedBy*”, “*FirstResponder*”, etc.) or to external domains (e.g., the domain of FOAF, OWL, etc.). However, at the same time such resources will not be accessible from outer domains.



Figure 6.7 Redirection to the restricted resources

Furthermore, the created certificates are used to restrict all resources belonging to the domain where the CF-CoC is residing. A certificate can be created not only for all resources on the server, but it can be issued for a specific resource on that server. For example, if there is a resource 'x' in DS1, then the field of the certificate called "issued to" (see Figure 6.3) will be assigned the complete URL of the resource 'x' (e.g., CN=Neutralside/resources/x).

6.5 Heartbleed: an error in the OpenSSL tool

An error should be highlighted and considered before ending this chapter related to the OpenSSL tool. This error is called the "heartbleed bug". This bug has been registered in the Common Vulnerabilities and Exposures system (CVE) as CVE-2014-0160 (Common Vulnerabilities and Exposures (CVE), 2014).

In december 2011, a developer working on the OpenSSL called "Robin Seggelmann"³² made a programming error in the module implementing the TLS heartbeat protocol in the OpenSSL 2012 version³³. This error is classified as a buffer over-read³⁴ (i.e., a situation where more data can be read than should be allowed), which allows hackers to get information transmitted by users from previous requests (e.g., those that are still stored in the temporary memory, session cookies, usernames and passwords, or sometimes may reveal the encryption keys). This error was considered a big gap in

³² <https://support.microsoft.com/en-us/kb/892987>

³³ <https://support.microsoft.com/en-us/kb/892987>

³⁴ <https://support.microsoft.com/en-us/kb/892987>

the module of heartbeat, that's why they called it heartbleed³⁵ (i.e., TLS heartbeat extension can be used to reveal up to 64k of memory to a connected client or server).

This error affected some websites and web-servers and allowed for two years to gain access to personal information on the web (i.e., those web sites and servers that used issued certificates between the period 2012 until April 2014, using this version).

This error was discovered in the late 2013 by two working teams from Google and the other from a Finnish company. They kept it secret until they re-implemented again the heartbeat module of the TLS.

Nowadays, this gap has been resolved in the new version of OpenSSL (1.0.1g). For those who used the old version, OpenSSL, they launched a command for upgrading the defective version to the new version 1.0.1g and issued new digital certificates using the corrected version. Today, the main concern of researchers is to know how many frauds are accomplished through this gap, which something difficult to reveal, because spying on the heartbeat does not leave any trace.

6.6 Conclusion

This chapter discussed in detail how the technology stack/linked data principles of the linked data are adapted to publish data into a closed scale while keeping the resolvability of these published resources. The idea is elaborated on the same case study provided in chapter 4 (i.e., same role players).

The represented resources are shared on closed scale between role players and the judge through the public-key infrastructure approach. This chapter opens the door to

³⁵ <https://support.microsoft.com/en-us/kb/892987>

a research representing the counter part of the LOD, called the LCD, which share all the advantages of the LOD, but with consumption restriction.

Therefore, the technology stack (URI, HTTP, and RDF) is enhanced to include a secure access mechanism (URI, HTTPS, and RDF). The work presented in this chapter is a bridge connecting dual works; the work proposed in (Cobden et al., 2011; Rajabi et al., 2012). In addition, it underlines that the digital certificates cannot be issued only for datasets, but also for resources within these datasets. Finally, this work also provides with technical details the complete scenario of how to use digital certificates to bend resources from LOD to LCD, in order to reach the compromise question between resolvability resources and their access restrictions.

CHAPTER VII

APPLYING THE CF-COC SYSTEM ON A COMPLETE FORENSIC PROCESS

7.1 Introduction

Chapter 1 formulated the hypotheses of this research. Chapter 2 collected the facts related to each problem and discussed them through different related works from literature. Chapter 3 analyzed and adapted such facts to reaching conclusions in the form of solutions toward the concerned problems. Chapter 4, 5, and 6 stimulated the production of desired information in the light of Chapter 3 and based on the facts discussed in Chapter 2.

The present chapter will discuss a complete experimentation to bring forth the desired information and improve the formulated hypotheses provided in chapter 1. This will be applied on a complete forensic model, to transform the tangible CoC associated to each phase into *e*-CoC, annotating it using different provenance metadata, and consume it on a closed scale by using different consumption patterns.

The same CoC provided in Chapter 3 and implemented along Chapter 4, 5, and 6, will be resumed to cover a complete forensic model in order to improve and validate the hypotheses proposed in chapter 1.

Referring to Chapter 2, different forensic models have been presented in Table 2.4. This chapter resumes and considers the Kruse model, because it encompasses the three essential forensic phases of any forensic investigation. This chapter will

consider the remaining tasks of the acquisition phase such as recovery and copy/backup. In the authentication phase, two tasks are considered to generate checksum and comparing them. Finally in the analysis phase, the backup device is examined in order to extract forensic information in the form accompanied with AFF4 format.

Based on the same footsteps of Figure 2.6, the Kruse model includes three forensic phases; each forensic phase contains a set of tasks.

“An IT company recalls a company specialized in cyber forensic to investigate and restore all excel sheet files from a PDA device. This device was using by an employee that has been fired recently from work and when he left, the IT company did not find those on his device. The employee was not authorized to delete or hide such type of files, because they are considered as intellectual sheets for this IT company. The cyber forensic company dedicated this investigation task to some role players called: Jean-Pierre, Peter, and Robert. Those role players applied the Kruse model in their investigation process. Jean-Pierre worked on preserving the device and did a classic recovery and restored some word document files. The excel files were not between such restored files. Jean-Pierre was not able to find the excel files, then he decided to escalate this issue to Pierre and Robert to make further investigations. Jean-Pierre then did a backup of the primary device on a secondary one. He provided both devices to Peter, who is an authenticator of digital media. Peter checked the integrity of both devices before analyzing of the secondary/backup device by Robert. After Peter finished checking the integrity, he provided the backup device to Robert. Robert in his turn analyzed the secondary device using a forensic tool called Encase. This forensic tool is able to detect and extract more and complicated information from the media device. He was able to notice through this tool that the size of the stored information is not reflecting its allocated size on the device. By using further investigations, he discovered that the fired employee used a program called secret

disk³⁶ to create a hidden partition within the current partition, where he hid all the excel sheet files.”

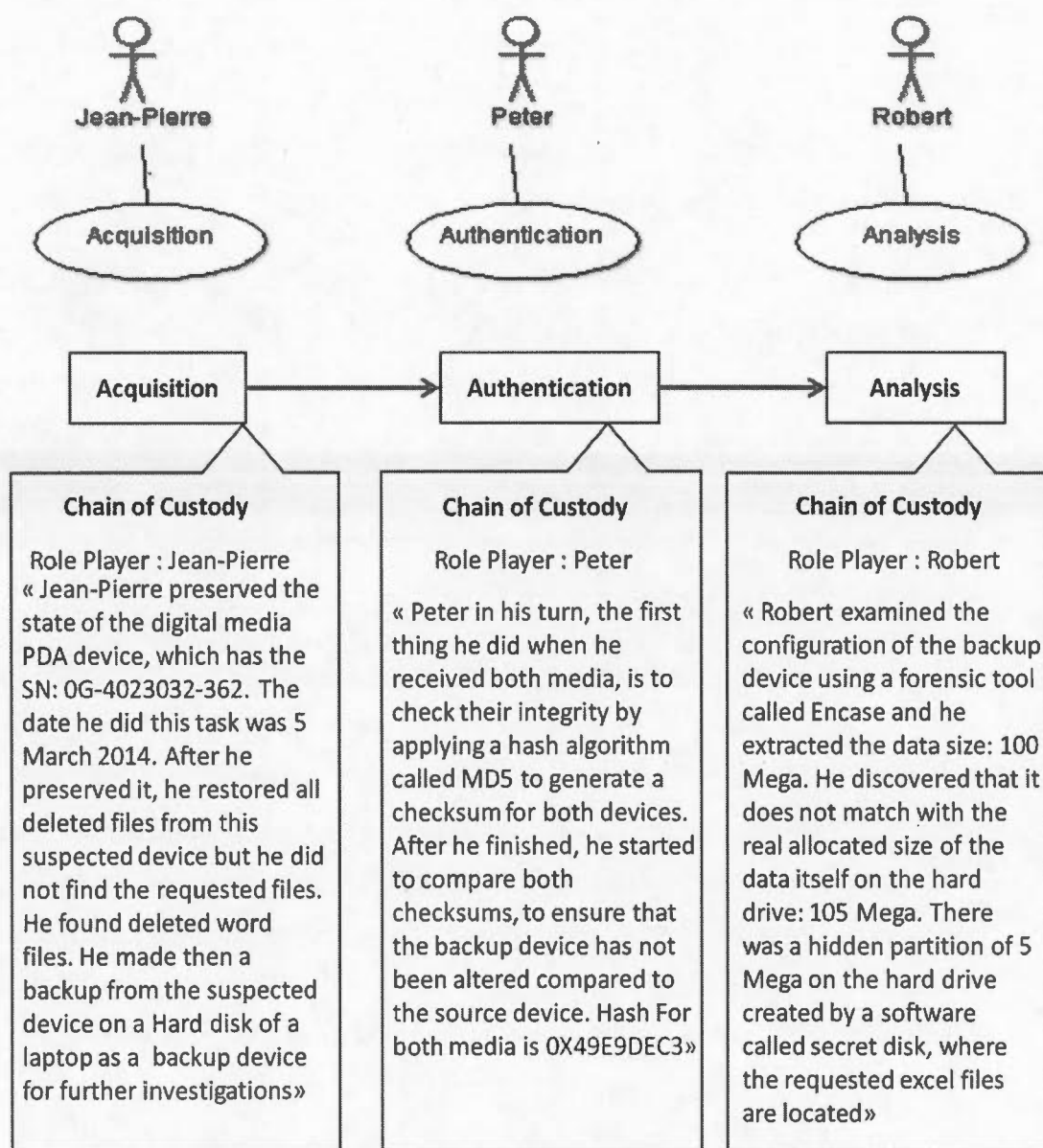


Figure 7.1

The tangible CoCs of the Kruse model

³⁶ <http://www.guidingtech.com/6765/hidden-files-invisible-partition-secret-disk/>

Figure 7.1 contains two parts; the upper part in the figure shows the use cases instances of the Kruse model and their role players. The lower part shows the corresponding CoC for each phase on the Kruse model that the role players will transform into *e*-CoCs using the CF-CoC system. Each phase containing a set of tasks. The acquisition phase contains preserve, recover, and backup. The authentication includes the generation of checksum for both primary and secondary device and then compares them. Analysis includes examining the secondary device.

Previous chapters (i.e., Chapter 4, 5, and 6), each one apart, explained the usage of CF-CoC system, how to transform the CoC into *e*-CoC, and annotate them, consume the published information using different consumption patterns, and bent such information on a closed scale. In this chapter, the complete picture combining all these tasks together is provided on a complete forensic process, in order to depict how the CF-CoC system proves all proposed hypothesis and depicts the solutions to answer research problems.

Next sections are presented based on the sequence of activities timeline. It starts by creating the client certificates for the technicians (i.e., by the CA through the neutral side. The technicians start to use CF-CoC to describe and publish their findings and annotate the published information using provenance metadata. Once the technicians finish their investigation and create the *e*-CoCs, they share the collected evidence and documents with the prosecutor. The latter communicates with defender and in case of dispute; the judge is engaged to get the testimony of the technicians. The example provided in this chapter will give a complete example of how technicians use the system to publish information (same idea for any other role players such as prosecutor and defender), and how the published information is consumed by the judge.

7.2 Identification of role players and judge

Before the forensic investigation starts, the issue and sharing of certificates takes place. Therefore, each party sends a list containing the name of their role players to the neutral side. The latter, in turn, sends both lists to the CA to issue their corresponding certificates. Also the neutral side asks from CA to issue a server certificate of the CF-CoC host and a client certificate for the judge.

Thus, for each role player there is a corresponding client certificate, one server certificate for the neutral side server, and a client certificate for judge. According to the use case mentioned above, five digital certificates will be issued by the CA (i.e., four client certificates and one server certificate). For client certificates: one for Jean-Pierre who is responsible to work on the acquisition phase, one for Peter who is responsible to work on the authentication phase, and the third is for Robert who is responsible to work on the analysis phase, and one client certificate for the judge. One server certificate issued for the neutral side, the owner of the CF-CoC host/serve. Finally, there is also a root certificate (i.e., of the CA itself) that is used to sign all certificates (i.e., clients and server).

After the CA issues all these certificates, they are all exported in *.p12* formats and sent to the neutral side to expedite to the role players and judge. The neutral side at this time is responsible to install its server certificate on the CF-CoC host. Once all certificates are issued and installed, each role player is able to start using the CF-CoC system to record the results of his forensic investigation.

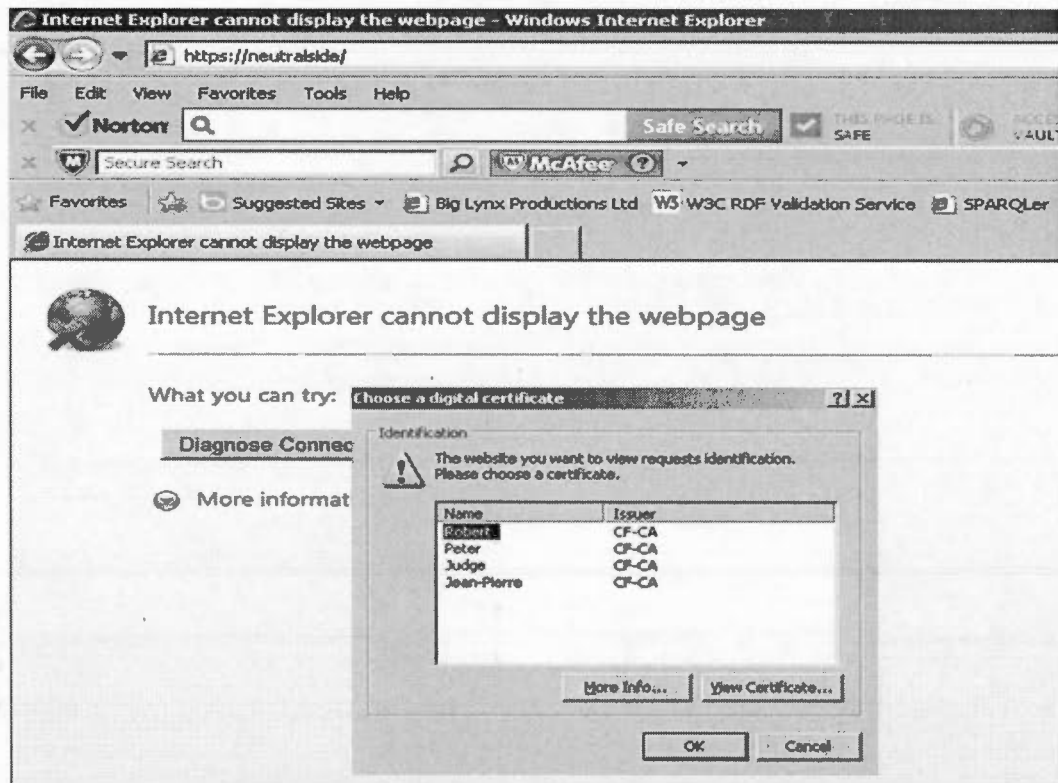


Figure 7.2 Screen showing some client certificates

The above figure, Figure 7.2, shows all the client certificates on the same screen, this is because all client certificates are tested and installed on only one machine, the fact that leads to show all the available certificates ready to communicate with the server certificate (i.e., they should have the same issuer, CF-CA) installed on the host of neutral side (i.e., in the figure below, it is the https://Neutralside). Once the server identifies the client certificate, it redirects the client (i.e., role player or judge) to the CF-CoC web application (see Figure 6.7). If a client certificate is not identified by the server certificate this will lead to situation illustrated in the Figure 6.6, and at this time neither the role players nor the judge will be able to use the CF-CoC to publish and secure the forensic information.

By using digital certificates, role players and judge are able to use the LDP, which are used to publish data publicly, to publish and consume forensic information on a closed scale. They answered the hypothesis #4 of securing the CoC information. Role players and judge are able through digital certificates to restrict the access to the forensic information while exploiting the advantages of using LDP for resolvability and representation, authenticate players to the neutral side server, and then restrict the consumption of information on a closed scale.

7.3 Publishing CoCs using CF-CoC

As mentioned in chapter 4, the role players start the publication task by selecting and determining which terms they are going to use, and by defining them using the vocabulary of the semantic web in order to record and publish forensic information.

As explained, the task of determining the forensic terms is subjective and differs from one role player to another. For example, as shown in the next table, the role player of the acquisition phase select the terms “*SuspectedDevice*” and “*SecondaryDevice*”, to refer to the primary and backup device. Other role players may use other terms to define both devices. Role players start to select the terms and then start to define them using the lightweight ontology vocabularies (i.e., RDFS++). Referring to the tangible documents provided in Figure 7.1, each role player defines his own terms (i.e., proprietary terms), their types (i.e., class, property), and assigns their constraints (i.e., tagged them using RDFS constructors) and use the CF-CoC to create them. Those custom terms are called proprietary terms.

In the next table (where “p” stands for a property and “C” stands for a class), each task in a forensic phase is described through a set of terms. This set of terms describes different pieces of forensic information. The main axis of each task is the

property term. For example, the recovery task is defined by the recover verb. The subjects and objects of each verb (i.e., each task), are described by domain and range classes.

Table 7.1 shows the term selected by the role players to create, record and publish their *e*-CoCs. For the acquisition phase, terms are mentioned in Table 4.1. The role player (*“FirstResponder”*) of this phase needs to determine more terms related to the other two forensic tasks (i.e., Recovery and Backup). For the recovery task, *“Jean-Pierre”* determined new terms. He determined new class term called *“SuspectedDevice”* beside the one *“DigitalMedia”* defined in the preservation task, and he defined the *“SuspectedDevice”* to be a subclass of the *“DigitalMedia”*. For the recovery task itself, he will define a property term called *“recover”*, where its domain is a subclass from *“Role player”* and its range will be a subclass of *“DigitalMedia”*. The thing that he is going to recover will be a set of deleted files. He will define a new class called *“DeletedFiles”*, which is a subclass of *“DeletedResource”*, which is a subclass of *“FileDataObject”* defined in an existing and well-defined vocabulary called NEPOMUK File Ontology (NFO) (ontology that deals with files and other desktop resources, whose super-class is the *“FileDataObject”* class that represents files from some digital storage medium)³⁷. *“Jean-Pierre”* defines a custom property called *“containsRecover”* as a sub-property of the property *“format”* imported from the Dublin (DCMI, 2015) that is used to describe either the physical or digital manifestation of a resource. He defined the domain class of this custom property to be the *“DigitalMedia”*, and its range will be a new custom class called *“DeletedFiles”*, which is a subclass of the class *“DeletedResource”* imported from the name space *“nfo”*. This means that Jean-Pierre needs to describe any digital media contains partitions and different files type.

³⁷ <http://www.semanticdesktop.org/ontologies/2007/03/22/nfo/>

Table 7.1 Forensic terms of Kruse model

Phase	Task	T-Box	A-Box
Acquisition	Preservation	RolePlayer (C), FirstResponder(C),DigitalMedia(C), SN(p), preservedBy (p), preserve(p)	Jean-Pierre, PDA device, LaptopHD, 0G4023-32-362, Word Files
	Recovery	recover (p), SuspectedDevice (C), containsRecover (p), DeletedFiles (C)	
	Copy/Backup	backup (p), BackupDevice (C), backupTo (p)	
Authentication	Generate Checksum	PrimaryDevice (C) + SecondaryDevice (C) + authenticatePrimary(P) + authenticateSecondary(p) + hashingPrimary (p) + hashingSecondary (p) + ImagefilePrimary (C) + ImagefileSecondary (C) checksumPrimary (p) + checksumSecondary (p) chckalgorithmPrimary (p) + chckalgorithmSecondary (p) + Authenticator (C)	Peter, Personal_Digital_ Assistant, Hard_disk_laptop HDL_image.img, PDA_image.img, MD5, 0X49E9DEC3,
	Compare	owl:sameAs	

Analysis	Analyze	Analyzer (C), analyze (p), analyzedBy (p), ForensicTool (C), dataSize (p), totalSize (p), HiddenPartition (C), contains (p), hiddenContains (p), hiddenUsing (p), hiddenSize (p)	Robert, hidden part, Excel files, Encase, 100 Mega, 5 Mega, 105 Mega, Secured disk
----------	---------	--	---

In the backup task, “*Jean-Pierre*” found that he needs to define more terms to describe this task. He defined a new term called “*BackupDevice*” which will be a subclass of the “*DigitalMedia*” class, and a “*copy*” property whose domain will be a subclass from the “*Role player*” class and range will be a subclass of “*DigitalMedia*”. He also defined a new property term called “*CopyTo*”, where its domain will be the source device (i.e., “*SuspectedDevice*”) and its range will be the destination device (i.e., “*BackupDevice*”).

The second forensic phase is the authentication one. In this phase, the role player “*Peter*” defined his role by using a class called “*Authenticator*”, which is a subclass of “*RolePlayer*”. He defined new terms for the suspected and backup devices. He did not notice terms defined by “*Jean-Pierre*”, he defined his own terms for both devices: “*PrimaryDevice*” and “*SecondaryDevice*”. In such case, this will not change anything if a role player reveals that both terms are referring to the same concept, a mapping using “*sameAs*” constructor can be used to relate both terms together. In this phase, there are two main forensic tasks to verify the integrity of both devices: generate the checksum for each device (i.e., suspected and backup devices), then compare both checksum to ensure that they are identical, which means that the backup device is not altered or tampered and consistent with the source device.

In generating the checksum, “*Peter*” is an authenticator, so he defined his role as a subclass from the “*RolePlayer*” class, which is defined by “*Jean-Pierre*” in the preservation task of the acquisition phase.

The role player “*Peter*” defined also two new property terms called “*authenticatePrimary*” and “*authenticateSecondary*”. Both are sub-properties of the “*made*” property imported from FOAF vocabulary. The domain and range of both of them are “*RolePlayer*” and “*FileDataObject*”, respectively. The “*FileDataObject*” class is also a sub-class of “*rdfs:Resource*”, which is the class of all resources.

The authenticator “*Peter*” also defined two new terms for the “*SuspectedDevice*” and “*BackupDevice*”, the “*PrimaryDevice*” for the former and “*SecondaryDevice*” for the later. He defined these two new terms instead to use the term “*DigitalMedia*”.

He defined the “*hashingPrimary*” for the “*PrimaryDevice*” and the “*hashingSecondary*” for the “*SecondaryDevice*”. The domain and range of each of them are sub-classes of “*FileDataObject*” being verified and “*ImagefilePrimary*” and “*ImagefileSecondary*” being generated, respectively

The authenticator “*Peter*” defined two classes for the image files the “*ImagefilePrimary*” and “*ImagefileSecondary*”. Each of them will be a sub-class of “*FileHash*” class defined in NEPOMUK File Ontology. The “*ImagefilePrimary*” / “*ImagefileSecondary*” is a custom class referring to the image file fingerprint generated from the hashing task and generating the checksum.

For the checksum algorithms “*Peter*” also defined “*chckalgorithmPrimary*” / “*chckalgorithmSecondary*” (property), which will be a sub-property of the property called “*hasAlgorithm*” defined in the NFO ontology. Its domain will be the image file “*ImagefilePrimary*” / “*ImagefileSecondary*” class generating from hashing and its range will be the name of this algorithm inherited from XML Schema Datatype (W3C, 2006).

Also, he defined a property term called “*checksumPrimary*” / “*checksumSecondary*” that shows the hashing value of the image fingerprint (i.e., known by checksum or hashing value). It will be a sub-property of “*hasValue*” property defined in the NEPOMUK File Ontology, with as domain “*ImagefilePrimary*” / “*ImagefileSecondary*” and as range the checksum string itself inherited from XML Schema Datatype. Finally, he will use the “*owl:sameAs*” to assert the checksum generated from both devices.

Finally, the analyzer “*Robert*” of the analysis forensic phase will use the same term used by “*Peter*” to identify the backup device (i.e.; secondary device). He also defined two properties “*analyze*” and “*analyzedBy*” property to describe who did what.

A “*ForensicTool*” class is also defined by “*Robert*” to refer to the tool used in the forensic investigation. A “*dataSize*” property term referring to the current allocation of data, “*totalSize*” property will be used to refer to the real size of data on the hard drive.

The analyzer “*Robert*” defined “*contains*” property that will be used to describe the content of any partition (not hidden). Also, he defined the class “*HiddenPartition*” to refer to hidden part of the hard drive and “*hiddenContains*” property to express the fact that this part contains hidden information. The tool used to hide this part on the hard disk will be explained through the term of “*hiddenUsing*” property, and finally the size of the hidden partition will be explained through the “*hiddenSize*” property. This will explain in detail in Section 7.3.3 of the analysis phase.

Next sections will explain the remaining tasks for the acquisition phase (i.e., recovery and backup) and the tasks of authentication and analysis. Preservation task has been discussed as an example along chapters 4, 5 and 6.

7.3.1 The acquisition phase

As explained, the acquisition phase contains three forensic tasks: the preservation of media, its recovery and its backup. Next sections discuss each section apart.

7.3.1.1 Recovery

The recovery task is the second task after the preservation task in the acquisition phase. In the recover task, Jean-Pierre starts to define the “*SuspectedDevice*” to be a subclass of the “*DigitalMedia*” defined under the preservation category task (see Figure 7.3).

Create New Forensic Term		
Term Name : *	SuspectedDevice (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input checked="" type="radio"/> New Recovery <input type="radio"/> Existing	
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
	From	Custom Ontology
<input checked="" type="checkbox"/> Subclass-of	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	DigitalMedia (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term Suspected device	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term The source device	
Create New Term		

Figure 7.3 Screen for creating “*SuspectedDevice*” class

By creating the new class term “*SuspectedDevice*”, the system shows its associated RDF model (see Figure 7.4):

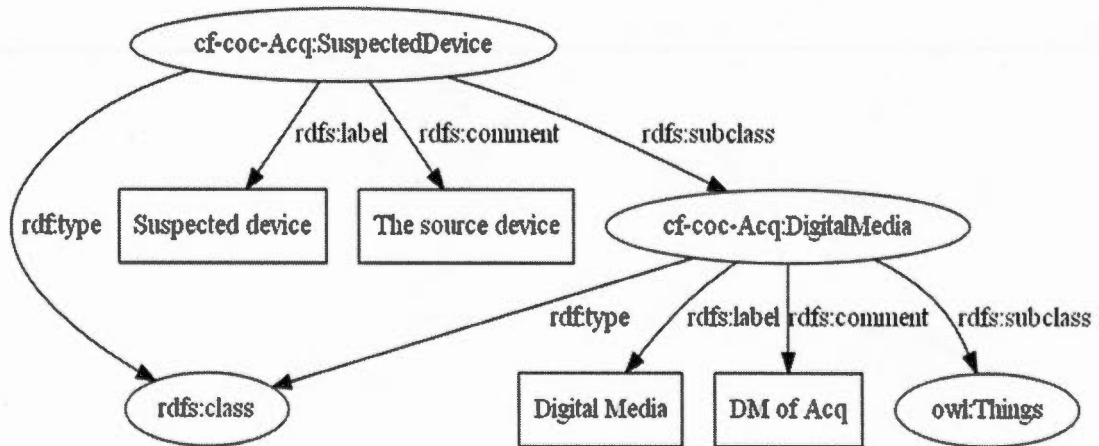


Figure 7.4 RDF Model for “*SuspectedDevice*” class

He also defined a property term “*recover*”, he tagged this term to be a “*FunctionalProperty*”, because in this task the role player recovers only one device, the source device. As shown in Figure 7.5, “*Jean-Pierre*” defined the recovery task in the acquisition phase and he defined this property as a sub-property of the property “*made*” defined in the FOAF namespace. He selected as range the “*SuspectedDevice*” that he came to define and the domain is the “*FirstResponder*”, which is a subclass of “*RolePlayer*”.

Term Name : *	recover (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	SuspectedDevice (Recovery)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	FirstResponder (Preservation)
<input type="checkbox"/> Label	Enter a label for the term. Because of deleted files	

Figure 7.5 Screen for creating “*recover*” property

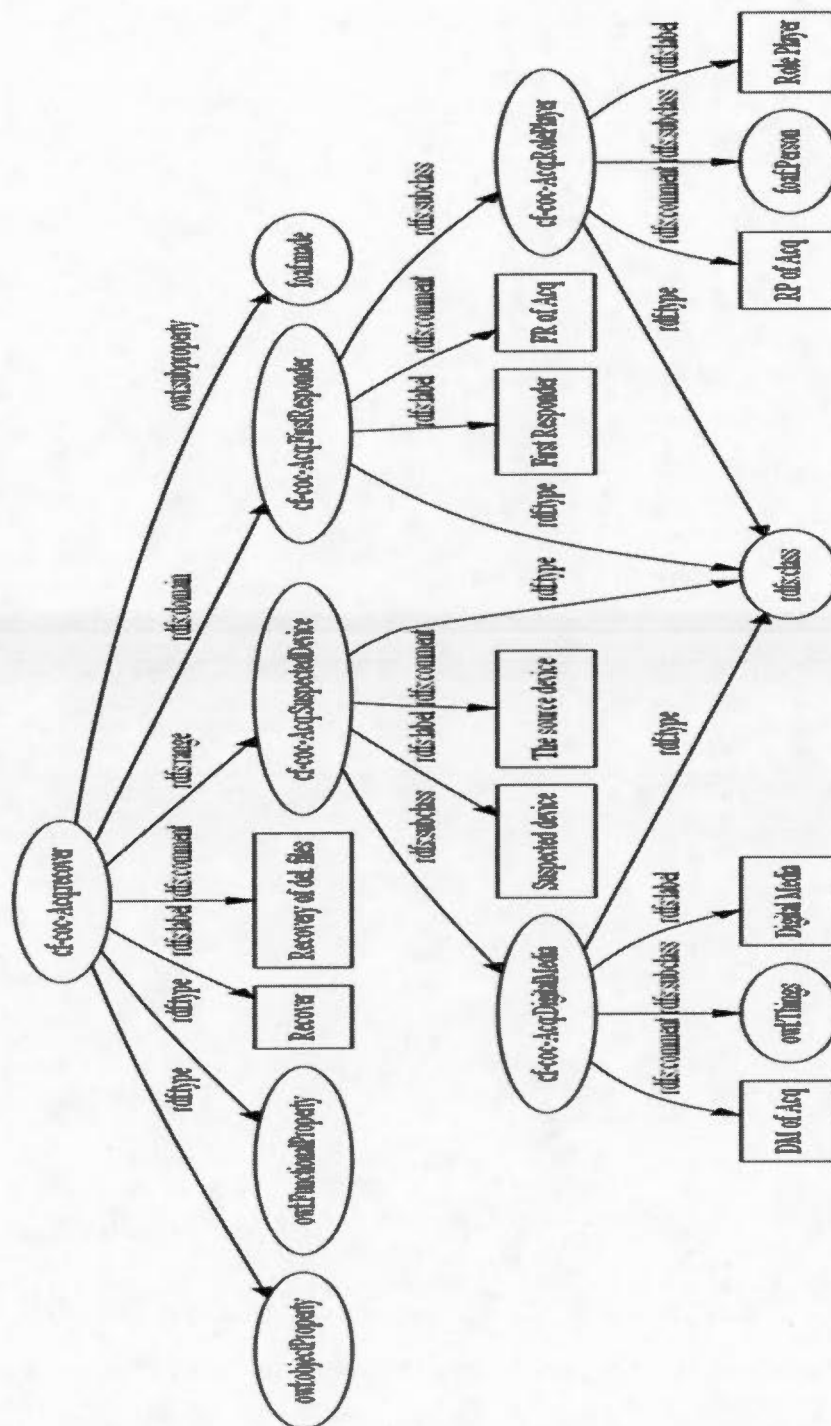


Figure 7.6 RDF Model for the “recover” property

By creating the new property term “*recover*”, the system shows the associated RDF model (see Figure 7.6).

Jean-Pierre defined also a class term called “*DeletedFiles*”, which is a subclass of the “*nfo:DeletedResource*” class (i.e., see Figure 7.7).

Create New Forensic Term

Term Name : *	DeletedFiles (Specify the name of the new term)		
In Ontology : *	Acquisition (Specify in which ontology you define a new term)		
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Acquisition	Recovery
Term Type : *	Class (Specify the type of the new term)		
RDF-Schema Vocabulary			
<input checked="" type="checkbox"/> Subclass-of	From	Built-in Ontology	
	Ontology Name	NEPOMUK_FILE_Ontology (nfo)	
	Class Name	DeletedResource (Files)	
<input checked="" type="checkbox"/> Label	Enter a label for the term deleted files		
<input checked="" type="checkbox"/> Comment	Enter a comment for the term files have been deleted		
Create New Term			

Figure 7.7 Screen for creating the “*DeletedFiles*” class

By creating the new class term “*DeletedFiles*”, the system shows the associated RDF model (see Figure 7.8):

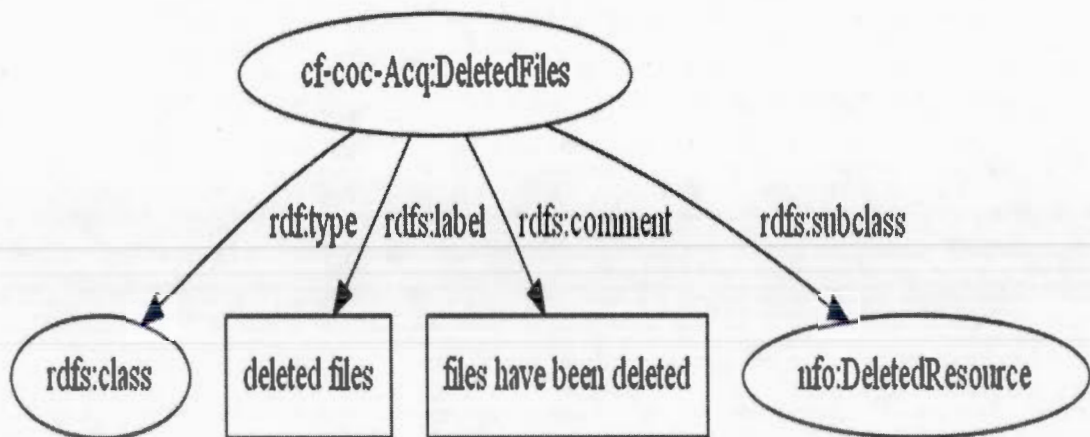


Figure 7.8 RDF Model for the “*DeletedFiles*” class

Jean-Pierre defined, as well, a new property term called “*containsRecover*” to describe the results of the recovered files from the “*SuspectedDevice*”. Thus, the domain of this property will be the “*SuspectedDevice*”, whereas its range will be the deleted files presented in the “*DeletedFiles*” class. Figure 7.9 indicates the definition of this property term:

Create New Forensic Term

Term Name : *	containsRecover (Specify the name of the new term)		
In Ontology : *	Acquisition (Specify in which ontology you define a new term)		
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Acquisition	Recovery
Term Type : *	Property (Specify the type of the new term)		
RDF-Schema Vocabulary			
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology	
	Ontology Name	Dublin_Core (dc)	
	Property Name	format (http://purl.org/dc/terms/format)	
<input checked="" type="checkbox"/> Range	From	Custom Ontology	
	Ontology Name	Acquisition (cf-coc-Acq)	
	Class Name	DeletedFiles (Recovery)	
<input checked="" type="checkbox"/> Domain	From	Custom Ontology	
	Ontology Name	Acquisition (cf-coc-Acq)	
	Class Name	SuspectedDevice (Recovery)	
<input checked="" type="checkbox"/> Label	Enter a label for the term results		

Figure 7.9 Screen for creating “*containsRecover*” property

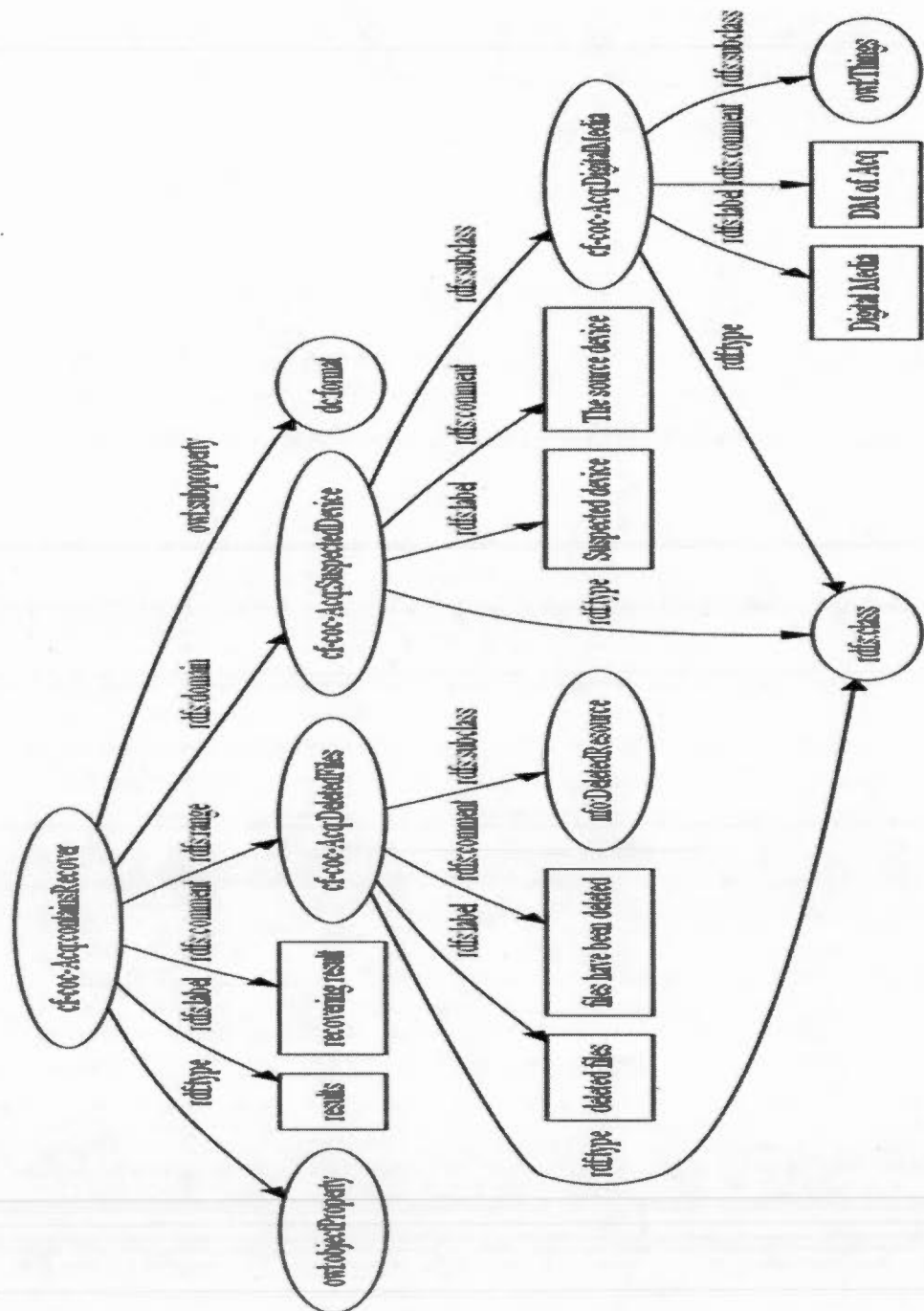


Figure 7.10 RDF Model for the “containsRecover” property

7.3.1.2 Backup

Backup is the third task of the acquisition phase, after preservation and recovery. In this task, the role player performs a backup on the suspected device by copying it to a backup device. Jean-Pierre, the role player of this phase, defines a property term describing the task itself, called “*backup*” whose domain will be the “*FirstResponder*” class defined in the preservation task and range will be the “*SuspectedDevice*”. The term of backup task “*backup*” is tagged also using the constructor “*FunctionalProperty*”, to restrict this task to only one source device. In this case we assume that the first responder makes a backup for only one suspected device during the investigation phase. This means that the backup device (i.e., primary/source/suspected) is the only device that obeys to this type of task (see Figure 7.11).

Term Name : *	backup (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
	Acquisition	Backup
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	SuspectedDevice (Recovery)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	FirstResponder (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term backup	

Figure 7.11 Screen for creating the “*backup*” property

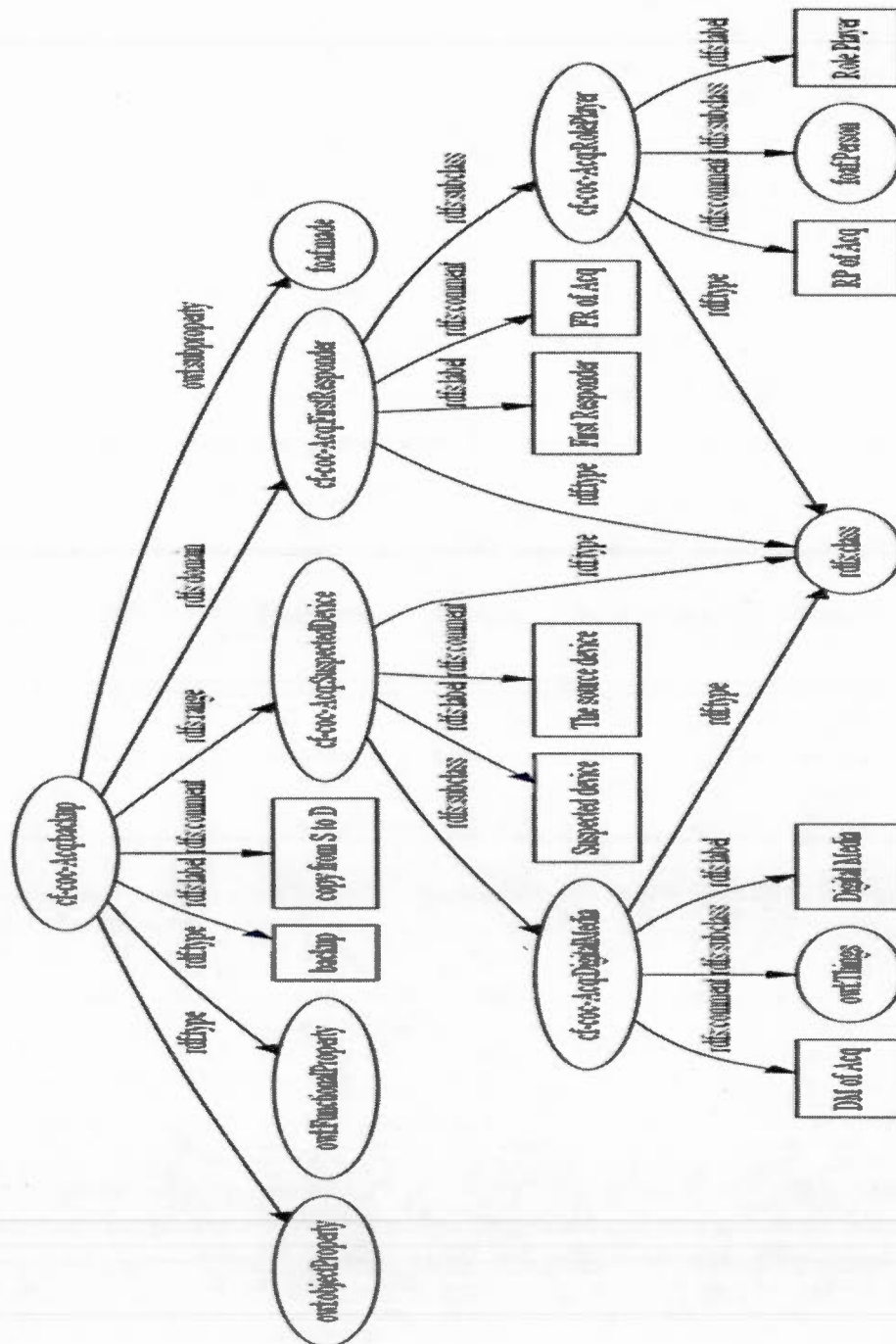


Figure 7.12 RDF model for the “backup” property

The last two terms that are defined by “Jean-Pierre” are the “*BackupDevice*” class and “*backupTo*” property. The definition of the “*BackupDevice*” class term is similar to the class “*SuspectedDevice*”. It is also a sub-class of the “*DigitalMedia*” class defined in the preservation task (See Figure 7.3 and 7.4).

The “*backupTo*” is a property used to explain what will be the source and the backup device in the backup task. Thus, the domain of this term will be the “*SuspectedDevice*” class and its range will be the “*BackupDevice*” (see Figure 7.13). This property is tagged using the constructors *owl:FunctionalProperty* and *owl:InverseFunctionalProperty*, to assess a one to one relationship (i.e., like the one to one cardinality). A source device can not be a backup on more than one device, and at the same time the backup device can not be a backup to more than one source device. It is a one to one relationship between both devices.

Term Name : *	backupTo (Specify the name of the new term)	
In Ontology : *	Acquisition (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Acquisition Backup
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	BackupDevice (Backup)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	SuspectedDevice (Recovery)
<input checked="" type="checkbox"/> Label	Enter a label for the term backup to	

Figure 7.13 Screen for creating the “*backupTo*” property

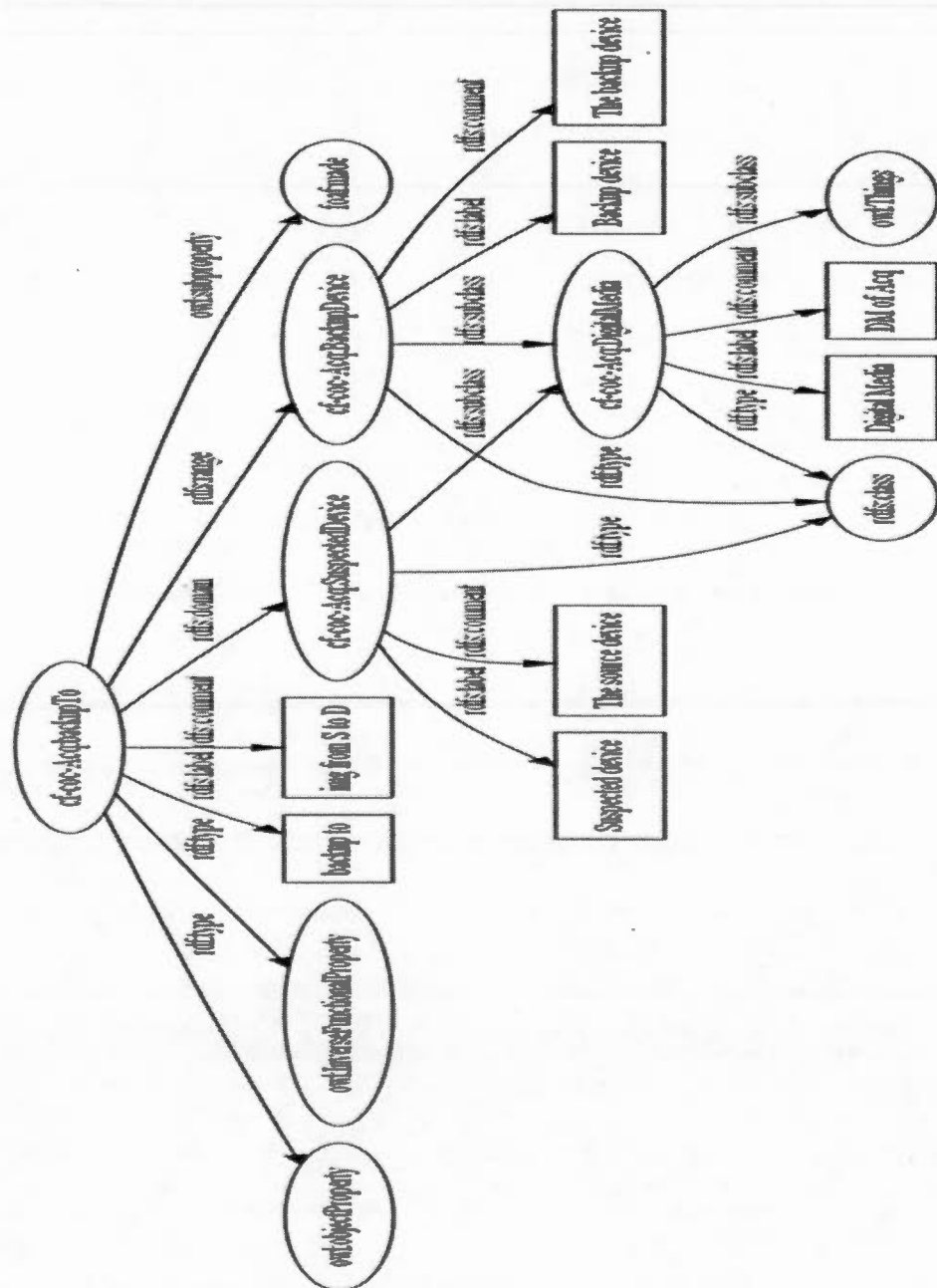


Figure 7.14 RDF model for the “*backupTo*” property

By creating the new property term “*backup*”, the system shows the associated RDF model (see Figure 7.12):

7.3.1.3 The e-CoC of the acquisition phase

After the definition of terms, Jean-Pierre starts using the defined terms to publish and generate the *e*-CoC of his forensic phase. What he did during his acquisition phase, aside from the preservation task explained in Chapter 4, 5 and 6 can be summarized into a set of triples:

1. Jean-Pierre *recover* PDA device
2. PDA device *containsRecover* wordfiles
3. Jean-Pierre *backup* PDA device
4. PDA device *backupTo* Harddisk

Next figure shows an example of how to publish and generate the second triple and underline two main characteristics. As explained before, the main vector to publish an RDF triple is the predicate slot. First, once the role player selects the predicate, the system advises him about the subject (i.e., domain) and object (i.e., range) of this predicate. In this figure, when the role player selects the “*containsRecover*” property, the systems shows that the domain of this property is the “*SuspectedDevice*” class and its range is the “*DeletedFiles*” class. Then the role player can select the subject that he is going to publish from the triple that has been defined before in the acquisition phase, or it will be a new resource to define. If it is a new resource then he will select the first option, if not, he will browse if he or another players already defined a resource by which he will publish his triple with the predicate in question. In the case of Figure 7.15, the role player selected in the subject slot, and he found that the PDA device was already created by himself during the preservation task (i.e., see Chapter 4), while in the object slot he defined a new literal or resource which will be an instance from the “*DeletedFiles*” class.

Both characteristics facilitate the publication task and help the role player avoid redundancy and later allow mapping between instances.

Publish RDF Triples

Subject	Predicate	Object
<input type="radio"/> New Resource <input type="radio"/> Existing Resource <input type="text" value="Acquisition"/> <input type="text" value="PDA device"/>	From <input type="text" value="Custom Ontology"/> Ontology Name <input type="text" value="Acquisition (cf-coc-Acq)"/> Property Name <input type="text" value="containsRecover (Recovery)"/>	<input type="radio"/> New Resource/Literal <input type="text" value="Word Files"/> <input type="text" value="Acquisition"/> <input type="radio"/> Existing Resource/Literal
Domain : SuspectedDevice		Range : DeletedFiles
<input type="button" value="Publish and Draw"/>		

Figure 7.15 Screen of publishing a triple using the “*containsRecover*” predicate

After publishing the above four triples, combined with those triples of the preservation task (see Chapter 4), the system generates the *e-CoC*, and its serialization code using RDF/XML (i.e., future option of the framework) of the acquisition phase (see Figure 7.16 and 7.17).

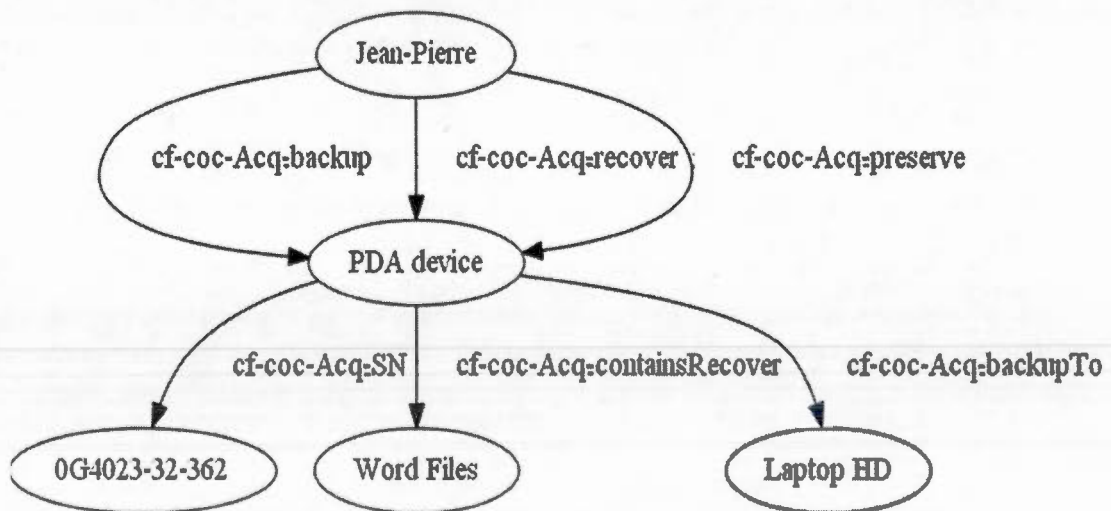


Figure 7.16 *e-CoC* of the acquisition phase


```

<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cf-coc-Acq=https://cyberforensics-coc/Acquisition/
xmlns:rdfs="http://www.w3.org/2000/10/XMLSchema#">
  <rdf:Description rdf:about="Jean-Pierre">
    <cf-coc-Acq:backup rdf:resource="PDA device"/>
    <cf-coc-Acq:recover rdf:resource="PDA device"/>
    <cf-coc-Acq:preserve rdf:resource="PDA device"/>
  </rdf:Description>
  <rdf:Description rdf:about="PDA device">
    <cf-coc-Acq:SN
rdf:datatype="http://www.w3.org/2000/10/XMLSchema#string">0G4023-
32-362
    </cf-coc-Acq:SN>
    cf-coc-Acq:containsRecover rdf:resource="Word Files"/>
    <cf-coc-Acq:backupTo rdf:resource="Laptop HD"/>
  </rdf:Description>
</rdf:RDF>

```

Figure 7.17 RDF/XML of the *e-CoC* of the acquisition phase

7.3.2 The authentication phase

The role player of this phase is “Peter”; he is the authenticator of the primary and secondary devices. An authenticator is responsible to check the integrity of both devices to ensure that they are not tampered. This phase contains two forensic tasks: generate checksum and compare checksum. First, he defined the ontology object of the authentication phase (i.e., container) to append all his custom terms to this object (i.e., same idea as Figure 4.2 and 4.3).

7.3.2.1 Generate checksum

First term, the role player Peter defined is the class term “Authenticator”. He defined this role to be a subclass from the “*RolePlayer*” class defined by Jean-Pierre in the acquisition phase (see Figure 7.18).

Term Name : *	Authenticator (Specify the name of the new term)	
In Ontology : *	Authentication (Specify in which ontology you define a new term)	
Category : *	<input checked="" type="radio"/> New <input type="radio"/> Existing	Generate Checksum
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
	From	Custom Ontology
<input checked="" type="checkbox"/> Subclass-of	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	RolePlayer (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term Authenticate	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term check integrity	
Create New Term		

Figure 7.18 Screen for creating the “Authenticator” class

By creating the new class term “Authenticator”, the system shows the associated RDF model (see Figure 7.19):

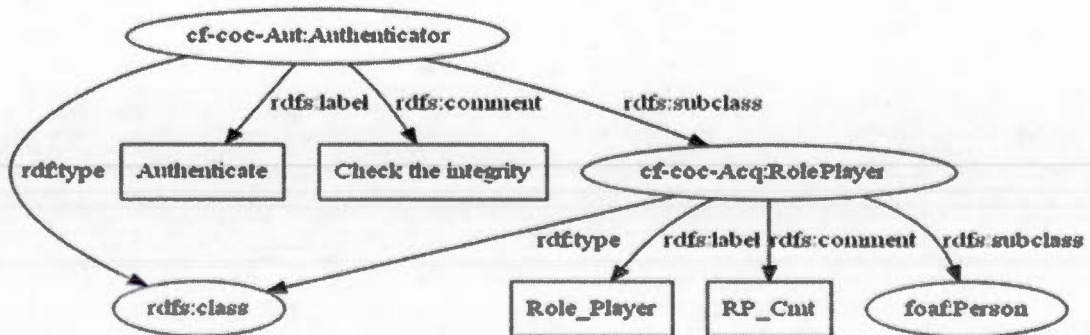


Figure 7.19 RDF model for the “Authenticator” class

Peter starts to create two classes to represent the primary and the secondary devices. He defines his own terms to refer to the original device and the backup device. From his point of view they will not inherit from the class “*DigitalMedia*” (i.e., which is a subclass of *owl:Things*), but he will define both terms using another way. He decides that they will be subclass from the “*FileDataObject*” class, which is a subclass of the class “*Resource*” defined in the RDFS vocabulary (see Figure 7.20).

Term Name : *	PrimaryDevice (Specify the name of the new term)		
In Ontology : *	Authentication (Specify in which ontology you define a new term)		
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Authentication	Generate Checksum
Term Type : *	Class (Specify the type of the new term)		
RDF-Schema Vocabulary			
<input checked="" type="checkbox"/> Subclass-of	From	Built-in Ontology	
	Ontology Name	NEPOMUK_FILE_Ontology (nfo)	
	Class Name	FileDataObject (Media)	
<input checked="" type="checkbox"/> Label	Enter a label for the term Source		
<input checked="" type="checkbox"/> Comment	Enter a comment for the term Source Dev		
Create New Term			

Figure 7.20 Screen for creating the “*PrimaryDevice*” class

By creating the new class term “*PrimaryDevice*”, the system shows the associated RDF model (see Figure 7.21). Figure 7.20 and Figure 7.21 can also illustrate the idea of creating a “*SecondaryDevice*”.

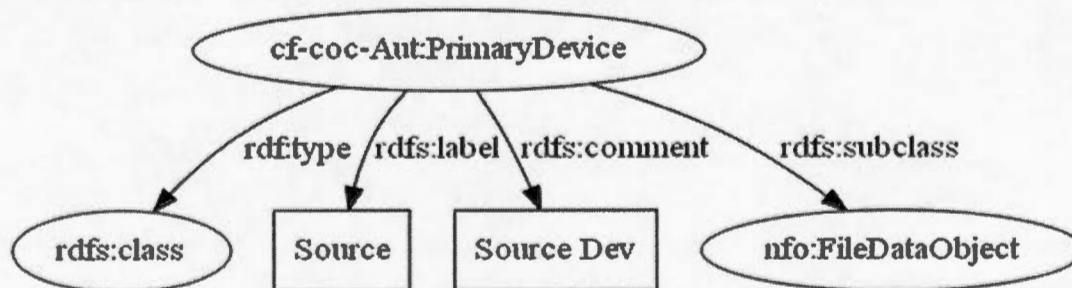


Figure 7.21 RDF model for the “*PrimaryDevice*” class

“Peter” wants to define the authenticate task itself. So he defines a new property term called “*authenticatePrimary*” and “*authenticateSecondary*”. He defines its domain to be “*Authenticator*” and the range to be whether the class “*PrimaryDevice*” and the class of the secondary device “*SecondaryDevice*”. Both are subclasses of the “*FileDataObject*” class (see Figure 7.22).

Term Name : *	authenticatePrimary (Specify the name of the new term)		
In Ontology : *	Authentication (Specify in which ontology you define a new term)		
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Authentication	Generate Checksum
Term Type : *	Property (Specify the type of the new term)		
RDF-Schema Vocabulary			
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology	
	Ontology Name	Friend_of_a_Friend (foaf)	
	Property Name	made (Documents and Images)	
<input checked="" type="checkbox"/> Range	From	Custom Ontology	
	Ontology Name	Authentication (cf-coc-Aut)	
	Class Name	PrimaryDevice (Generate Checksum)	
<input checked="" type="checkbox"/> Domain	From	Custom Ontology	
	Ontology Name	Authentication (cf-coc-Aut)	
	Class Name	Authenticator (Generate Checksum)	

Figure 7.22 Screen for creating the “*authenticatePrimary*” property

By creating the new property term “*authenticatePrimary*”, the system shows the associated RDF model (see Figure 7.23):

Peter defines also a property term for the hashing process, one for hashing the primary device using “*hashingPrimary*” and the second for hashing the secondary device using “*hashingSecondary*”. It concerns generating image fingerprint files for both devices. The domain of these properties will be “*PrimaryDevice*”, and “*SecondaryDevice*”, respectively. The range will be two new custom classes named “*ImagefilePrimary*” and “*ImagefileSecondary*”, which are subclasses of a well-defined class in the NFO called “*FileHash*” (see Figure 7.24). The same idea is applied, as shown below, for the secondary device.

Before explaining the “*hashingPrimary*” (i.e., same case for “*hashingSecondary*”), next figures (Figure 7.24 and 7.25) show how to define the “*ImagefilePrimary*” (i.e., same case for “*ImagefileSecondary*”).

Term Name : *	ImagefilePrimary (Specify the name of the new term)	
In Ontology : *	Authentication ▼ (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing <div>Authentication ▼ Generate Checksum ▼</div>	
Term Type : *	Class ▼ (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From	Built-in Ontology ▼
	Ontology Name	NEPOMUK_FILE_Ontology (nfo) ▼
	Class Name	FileHash (Hash) ▼
<input checked="" type="checkbox"/> Label	Enter a label for the term ImagefilePr.	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term fingerprint of S	
Create New Term		

Figure 7.24 Screen for creating the “*ImagefilePrimary*” class

The RDF model of Figure 7.24 is shown below:

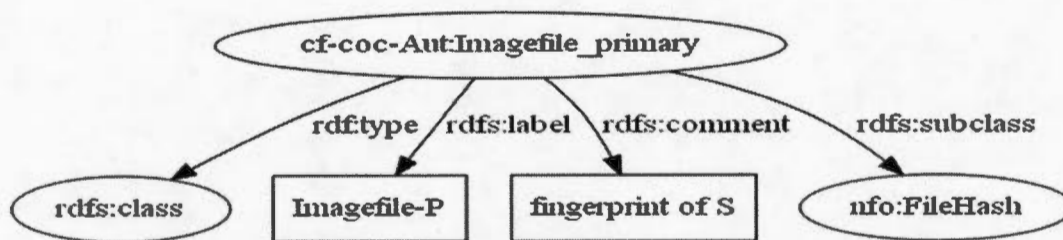


Figure 7.25 RDF Model for the “*ImagefilePrimary*” class

The next figure shows the definition of the hashing process. It shows the “*hashingPrimary*”. The same idea shown below for the primary device is also applied for the secondary device for “*hashingSecondary*”. This property is tagged to a Functional and *InverseFunctionalProperty*, since each media device after hashing has one and only one image file (*.img) and any image file is generating from only one device at a time.

Term Name : *	hashingPrimary (Specify the name of the new term)	
In Ontology : *	Authentication (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	NEPOMUK_FILE_Ontology (nfo)
	Property Name	hasHash (Hash)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	ImagefilePrimary (Generate Checksum)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	PrimaryDevice (Generate Checksum)
<input checked="" type="checkbox"/> Label	Enter a label for the term hashing	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term hash process	

Figure 7.26 Screen for creating the “*hashingPrimary*” property

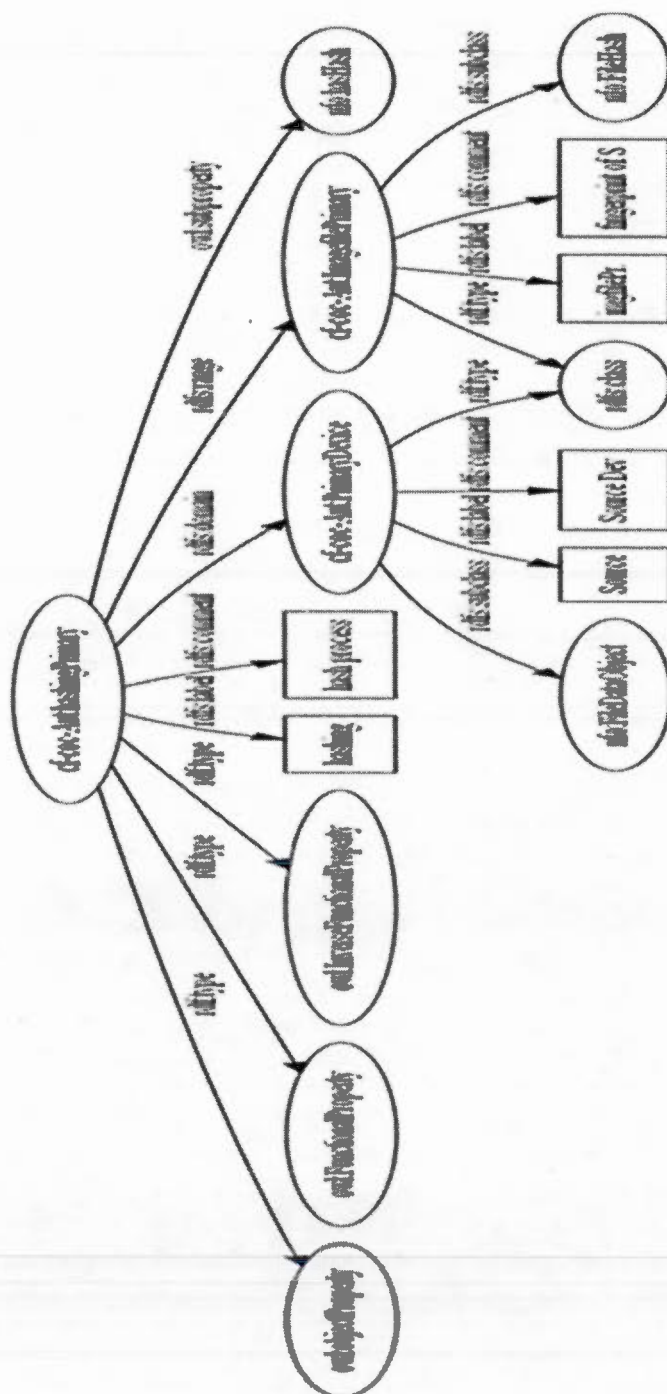


Figure 7.27 RDF model for “hashingPrimary” property

The hashing algorithm generates also a checksum string from the fingerprint image file. This is considered the value of the hashing process. Peter creates a new property term to define the checksum value. This term is called “*checksum*”, which is a sub-property of a well-defined property called “*hasValue*” in NEPOMUK File. Its domain will be the class of “*ImagefilePrimary/secondary*” and its range will be a string class for characters imported from the vocabulary XSL Schema (W3C, 2006). Figure 7.28 and 7.29 shows the definition of the checksum property for the primary device. The same idea is also applied to the checksum property for the secondary device.

The “*checksumPrimary*” property is also tagged to “*FunctionalProperty*” and “*InverseFunctionalProperty*”, because each fingerprint image file has a unique checksum value, and this unique value could not be the same for any other device using the hash algorithm (see Figure 7.28 and Figure 7.29).

Term Name : *	checksumPrimary (Specify the name of the new term)	
In Ontology : *	Authentication (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Authentication (Specify the type of the new term) Generate Checksum
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	NEPOMUK_FILE_Ontology (nfo)
	Property Name	hasValue (hash)
<input checked="" type="checkbox"/> Range	From	Built-in Ontology
	Ontology Name	XML_Schema (xsd)
	Class Name	String (string)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	ImagefilePrimary (Generate Checksum)
<input checked="" type="checkbox"/> Label	Enter a label for the term checksum	

Figure 7.28 Screen for creating the “*checksumPrimary*” property

The hash algorithm used to generate the checksum value is also defined using a property term called “*chckalgorithmPrimary*”. This term will be a sub-property of a well-defined property term called “*hasAlgorithm*”, with as domain class, image file generating from the hashing process and as range, the string class for characters imported from the vocabulary XSL Schema (see Figure 7.30 and 7.31) (W3C, 2006). The same idea will be applied to the property term “*chckalogrithmSecondary*”.

Term Name : *	chckalgorithmPrimary (Specify the name of the new term)	
In Ontology : *	Authentication (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Authentication <input type="button" value="Generate Checksum"/>
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From Built-in Ontology Ontology Name NEPOMUK_FILE_Ontology (nfo) Property Name hasAlgorithm (Hash)	
<input checked="" type="checkbox"/> Range	From Built-in Ontology Ontology Name XML_Schema (xsd) Class Name String (string)	
<input checked="" type="checkbox"/> Domain	From Custom Ontology Ontology Name Authentication (cf-coc-Aut) Class Name ImagefilePrimary (Generate Checksum)	

Figure 7.30 Screen for creating the “*chckalgorithmPrimary*” property

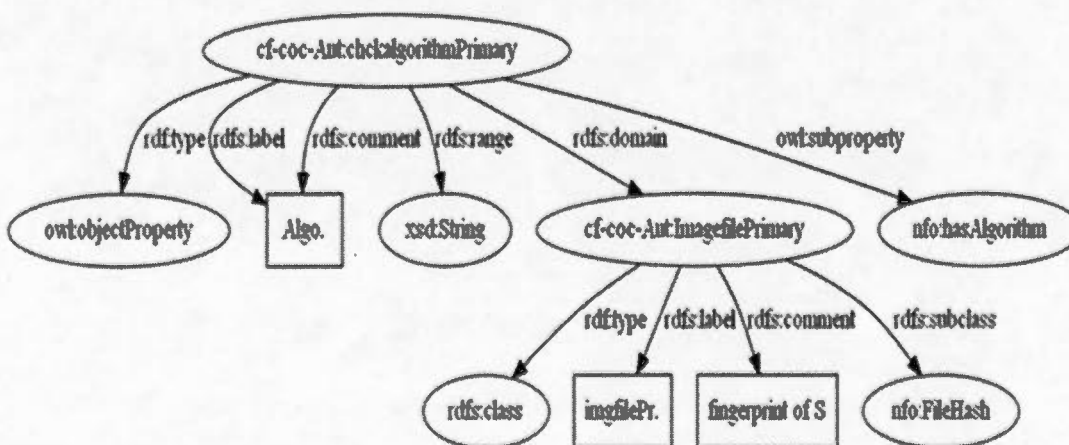


Figure 7.31 RDF model for the “*chckalgorithmPrimary*” property

7.3.2.2 Compare checksum

After generating the checksum for the primary and secondary devices, Peter compares both values and he finds that they are the same. This means that since the backup task took place until they are received by Peter, the backup device conforms and abides its integrity with the primary device. Peter does not define new terms and prefers to use the well-defined term “*owl:sameAs*” to refer that both values are the same.

7.3.2.3 The *e*-CoC of the authentication phase

After the definition of terms, Peter starts using these terms to publish and generate the *e*-CoC of his forensic phase. What he did during his authentication phase can be summarized into a set of triples:

1. Peter *authenticatePrimary* Personal_Digital_Assistant
2. Peter *authenticateSecondary* Hard_drive_laptop
3. Personal_Digital_Assistant *hashingPrimary* PDA_image.img
4. Hard_drive_laptop *hashingSecondary* HDL_image.img
5. PDA_image.img *checksumPrimary* 0X49E9DEC3
6. HDL_image.img *checksumSecondary* 0X49E9DEC3
7. PDA_image.img *chckalgorithmPrimary* MD5
8. HDL_image.img *chckalgorithmPrimary* MD5
9. 0X49E9DEC3 *owl:sameAs* 0X49E9DEC3

The following figure shows the *e*-CoC of the authentication phase.

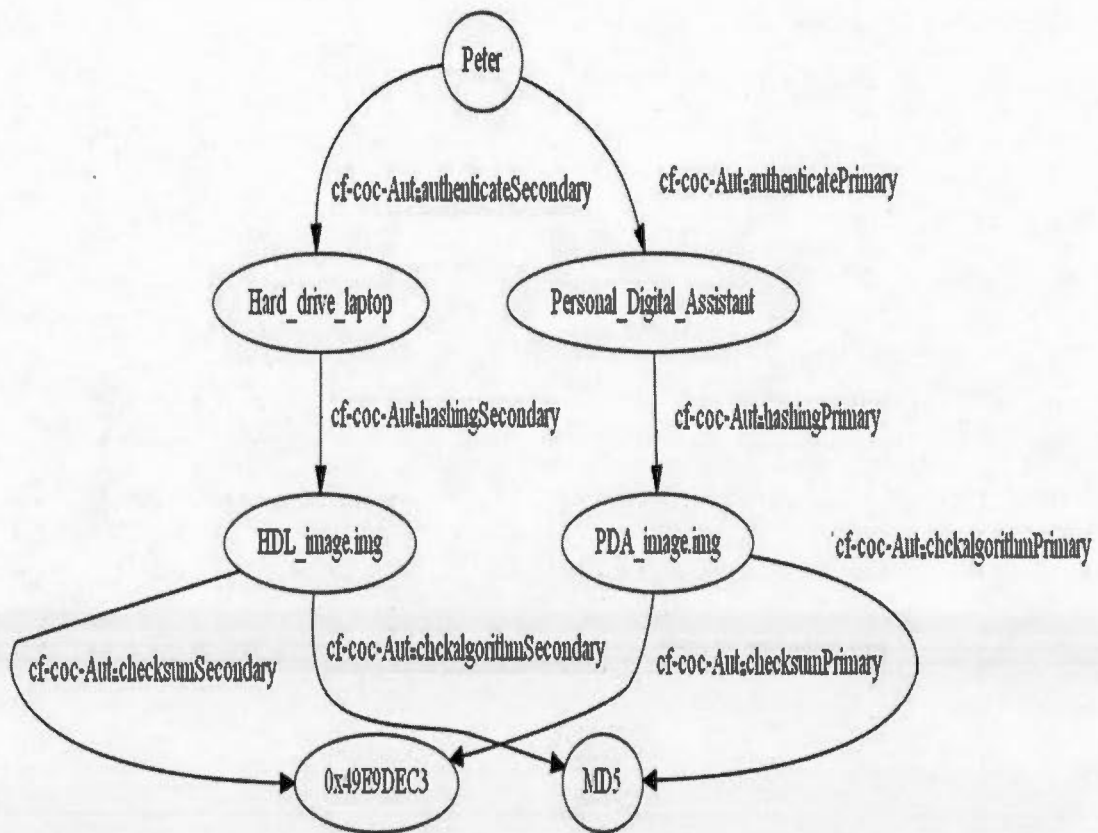


Figure 7.32 *e-CoC of the authentication phase*

As mentioned earlier, Peter defines his own terms for the primary and backup devices. This case may happen on the web of data, where a publisher may redefine concepts already defined by other publishers, because on the web of data it is unrealistic to assume that everybody will use the same name to refer to a certain concept. The mapping between terms can be performed using OWL constructor “*sameAs*”. The role players are responsible to relate these terms and the system can guide the role players through the reasoning pattern of the consumption module. Next figures show (Figure 7.33 and Figure 7.34), how the “*Personal_Digital_Assistant*” from the authentication phase is mapped into the term “*PDA device*” of the acquisition phase, and how the “*Hard_drive_laptop*” is related to the “*LaptopHD*” in the acquisition phase:

Publish RDF Triples

Subject	Predicate	Object
<input type="radio"/> New Resource	From <input type="text" value="Built-in Ontology"/>	<input type="radio"/> New Resource/Literal
<input checked="" type="radio"/> Existing Resource <input type="text" value="Acquisition"/> <input type="text" value="PDA device"/>	Ontology Name <input type="text" value="Ontology_Web_Language (owl)"/> Property Name <input type="text" value="sameAs (Owl Semantics)"/>	<input type="radio"/> Existing Resource/Literal <input type="text" value="Authentication"/> <input type="text" value="Personal_Digital_Assistant"/>
Refer to Vocabulary to get Property Domain		Refer to Vocabulary to get Property Range
<input type="button" value="Publish and Draw"/>		

Figure 7.33 Screen for mapping the source device

Publish RDF Triples

Subject	Predicate	Object
<input type="radio"/> New Resource	From <input type="text" value="Built-in Ontology"/>	<input type="radio"/> New Resource/Literal
<input checked="" type="radio"/> Existing Resource <input type="text" value="Acquisition"/> <input type="text" value="LaptopHD"/>	Ontology Name <input type="text" value="Ontology_Web_Language (owl)"/> Property Name <input type="text" value="sameas (Owl Semantics)"/>	<input type="radio"/> Existing Resource/Literal <input type="text" value="Authentication"/> <input type="text" value="Hard_drive_laptop"/>
Refer to Vocabulary to get Property Domain		Refer to Vocabulary to get Property Range
<input type="button" value="Publish and Draw"/>		

Figure 7.34 Screen for mapping the backup device

Combining the acquisition and authentication phase together, we will obtain the RDF model shown in Figure 7.35.

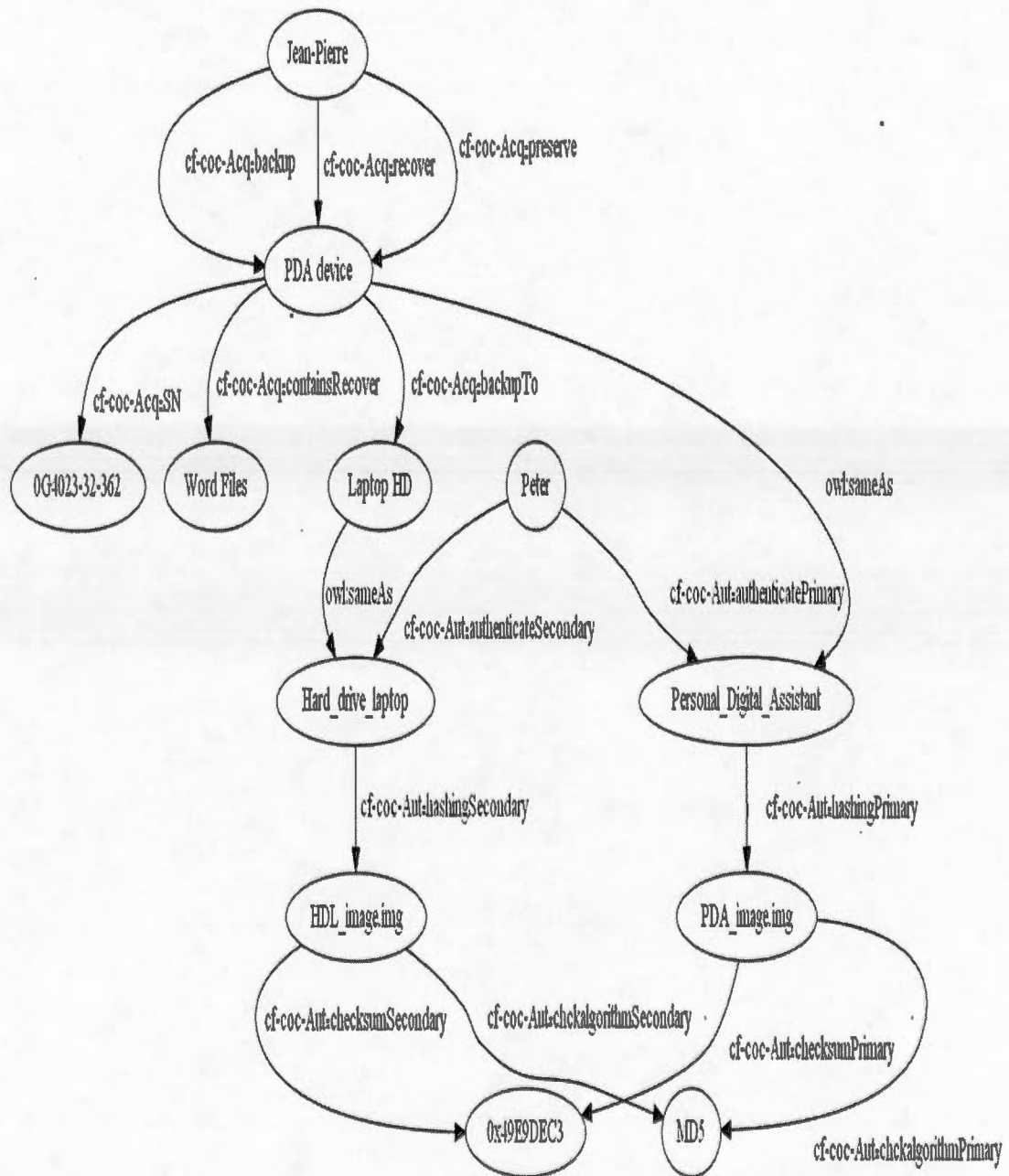


Figure 7.35 *e-CoC of acquisition and authentication phases*

7.3.3 The analysis phase

The role player of the analysis phase is Robert. He is the analyzer of the backup device sent by Peter, to examine it after ensured from its integrity with the source device. First, he defines the ontology object of the analysis phase (i.e., container) to append all his custom terms to this object (i.e., same idea as Figures 4.2 and 4.3).

7.3.3.1 Analyze

In the analyze task, first term he defines the “*Analyzer*” class, which will be a subclass of the “*RolePlayer*” class (i.e., same idea of Figure 7.18. See also Figure 7.36).

Term Name : *	Analyzer (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input checked="" type="radio"/> New <input type="radio"/> Existing	Analyze
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From	Custom Ontology
	Ontology Name	Acquisition (cf-coc-Acq)
	Class Name	RolePlayer (Preservation)
<input checked="" type="checkbox"/> Label	Enter a label for the term Analysis	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term Analyze the backup	
Create New Term		

Figure 7.36 Screen for creating the “*Analyzer*” class

By creating the new class term “*Analyzer*”, the system shows the associated RDF model (see Figure 7.37):

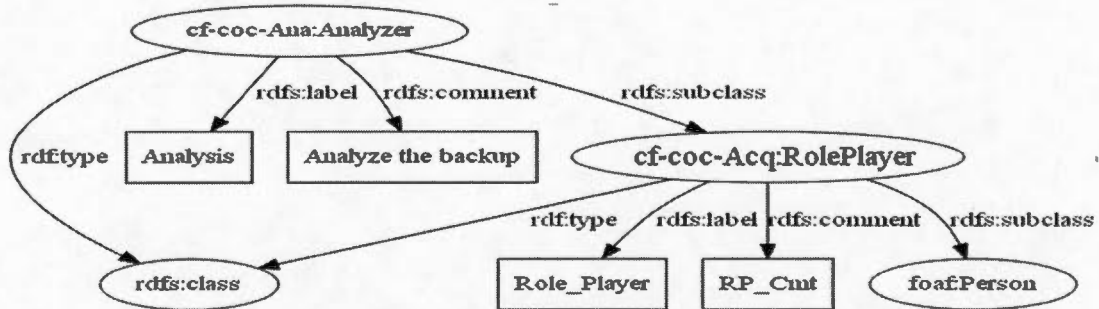


Figure 7.37 RDF model of the “*Analyzer*” class

Then, he defines the property of the analyze task, where its domain will be the “*Analyzer*” class and its range will be “*SecondaryDevice*” (i.e., he will use the same class defined by Peter in the authentication phase). He also tags this property using “*FunctionalProperty*”. Thus, all objects defined in a triple where “*analyze*” is the property then they will be considered as same object. See Figure 7.38.

Term Name : *	analyze (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	SecondaryDevice (Generate Checksum)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Analysis (cf-coc-Ana)
	Class Name	Analyzer (Analyze)
<input checked="" type="checkbox"/> Label	Enter a label for the term Analyze	

Figure 7.38 Screen for creating the “*analyze*” property

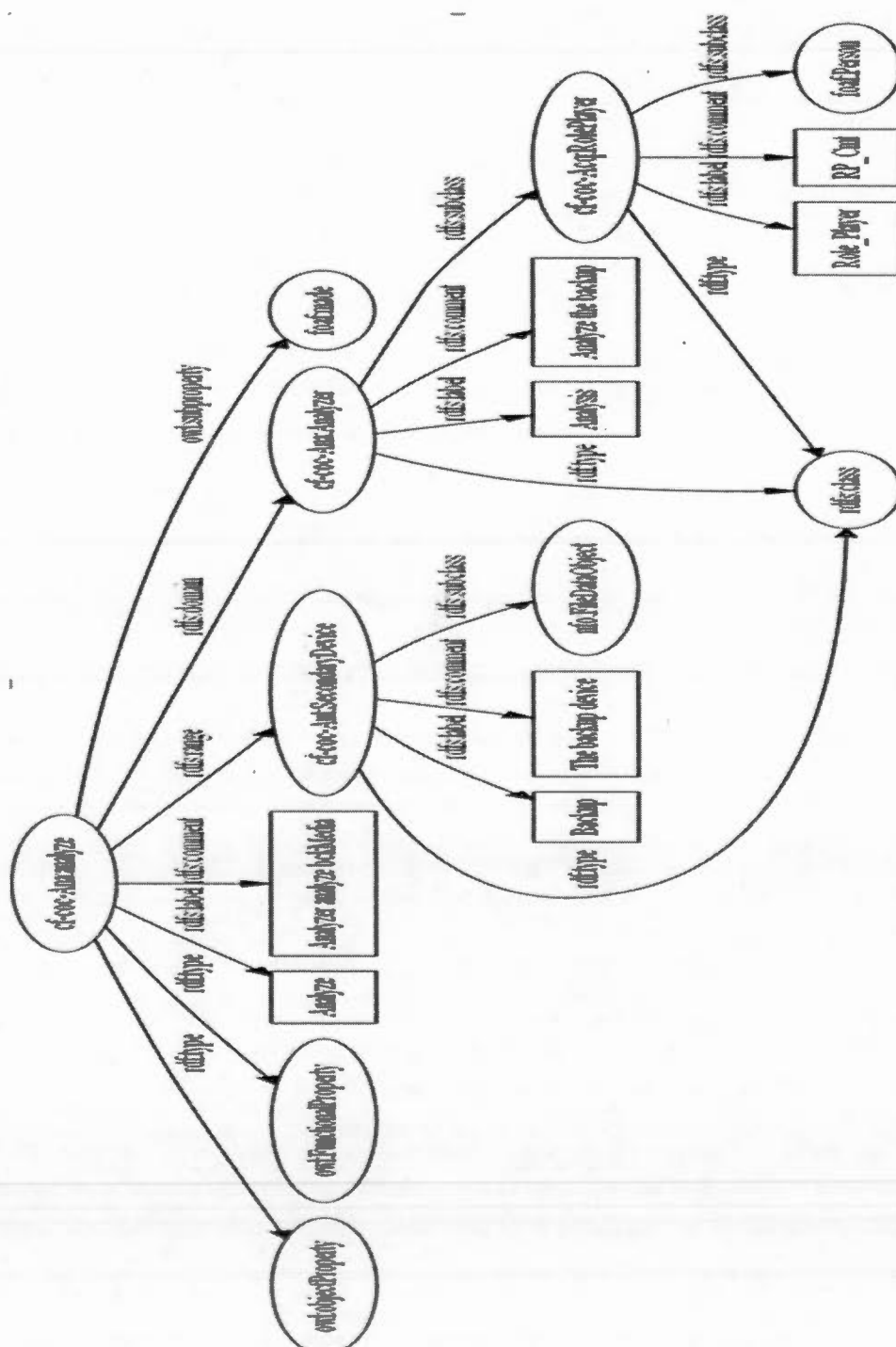


Figure 7.39 RDF model of the “analyze” property

As mentioned in Section 7.2, Robert will use the same term used by Peter to represent the backup device “*BackupDevice*”. He defines a new property called “*dataSize*” (i.e., to refer to the data size 100 Mega), which will be a sub-property of “*nie:contentSize*” defined within the Nepomuk Information Element Ontology (NIE). Its domain is the “*SecondaryDevice*” and its range is the “*xsd:integer*” (see Figure 7.40 and its RDF model in Figure 7.41).

Term Name : *	dataSize (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Analysis <input type="text"/> Analyze <input type="text"/>
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology <input type="text"/>
	Ontology Name	NEPOMUK_Information_Element (nie) <input type="text"/>
	Property Name	contentSize (Size) <input type="text"/>
<input checked="" type="checkbox"/> Range	From	Built-in Ontology <input type="text"/>
	Ontology Name	XML_Schema (xsd) <input type="text"/>
	Class Name	Integer (integer) <input type="text"/>
<input checked="" type="checkbox"/> Domain	From	Custom Ontology <input type="text"/>
	Ontology Name	Authentication (cf-coc-Aut) <input type="text"/>
	Class Name	SecondaryDevice (Generate Checksum) <input type="text"/>
<input checked="" type="checkbox"/> Label	Enter a label for the term dataSize <input type="text"/>	

Figure 7.40 Screen for creating the “*dataSize*” property

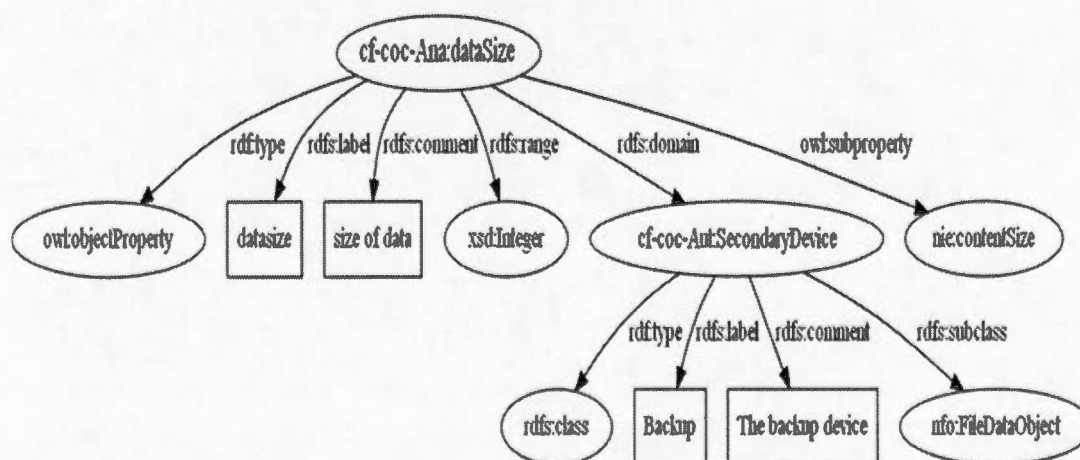


Figure 7.41 RDF model of the “*dataSize*” property

He defines a class called “*ForensicTool*”, which is a subclass of the class “*Software*” defined in the NFO. Also, he defines a property term called “*analyzedBy*”, which will be a sub-property of the *foaf:made*, and its domain will be the “*SecondaryDevice*” and its range will be “*ForensicTool*”.

Term Name : *	analyzedBy (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing Analysis Analyze	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Analysis (cf-coc-Ana)
	Class Name	ForensicTool (Analyze)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	SecondaryDevice (Generate Checksum)
<input checked="" type="checkbox"/> Label	Enter a label for the term Analyzed by	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term SW used to analyze	

Figure 7.42 Screen for creating the “*analyzedBy*” property

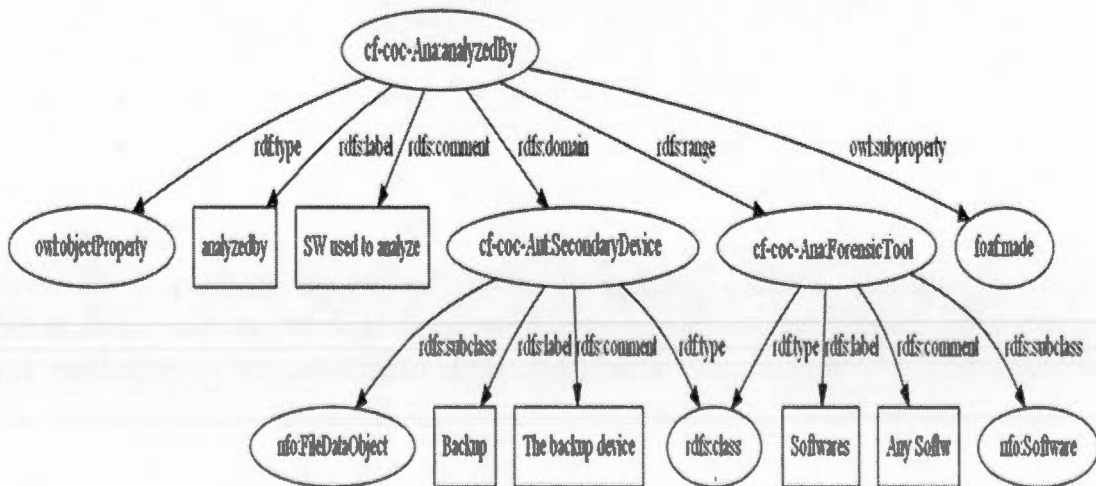


Figure 7.43 RDF model of the “*analyzedBy*” property

Robert also defines a new term property called “*totalSize*” (see Figure 7.44) to describe the real size of data (i.e., In our case, the hidden partition is 5 mega, and the unhidden is 100 mega, so the real total size of data will be 105 mega, see Figure 7.1). The property “*totalSize*” is the same idea of the “*dataSize*” property. It will also be sub-property of the “*nie:contentSize*”. Its domain will be the “*SecondaryDevice*” and its range is “*xsd:integer*”.

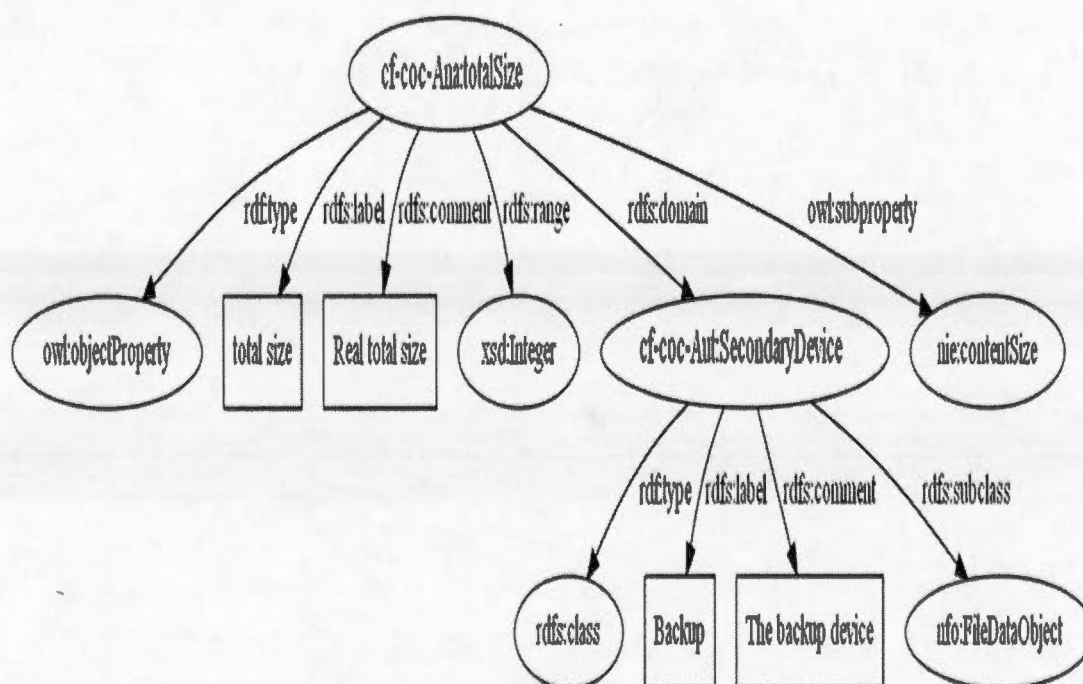


Figure 7.44 RDF model of the “*totalSize*” property

On the other hand, he defines a new property term called “*hiddenSize*” to refer to the size of the hidden partition in the backup device. This property will have a domain class referring to the hidden partition and its range will be the size of this hidden partition (5 Mega).

Robert defines as well, a new property terms called “*contains*”. He will use it to state that the backup device has a hidden partition created by the fired employee. This

property will be a sub-property of the property term “*hasPart*” defined within the Dublin Core vocabulary. The domain of this new property term will be “*SecondaryDevice*” defined by Peter in the authentication phase, and its range will be the hidden partition that he found on the hard disk using the Encase forensic tool. But before defining the property term “*contains*”, Robert defined the “*HiddenPartition*” class.

The “*HiddenPartition*” class will be a subclass of a well-defined class term in the NFO ontology called “*HardDiskPartition*” (see Figure 7.45).

Term Name : *	HiddenPartition (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	Analysis <input type="button" value="Analyze"/>
Term Type : *	Class (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subclass-of	From <input type="button" value="Built-in Ontology"/> Ontology Name <input type="button" value="NEPOMUK_FILE_Ontology (nfo)"/> Class Name <input type="button" value="HardDiskPartition (Partition)"/>	
<input checked="" type="checkbox"/> Label	Enter a label for the term <input type="text" value="Hidden Partition"/>	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term <input type="text" value="Part hidden in HD"/>	
<input type="button" value="Create New Term"/>		

Figure 7.45 Screen for creating the “*HiddenPartition*” class

Figure 7.46 shows the RDF model of the “*HiddenPartition*” class.

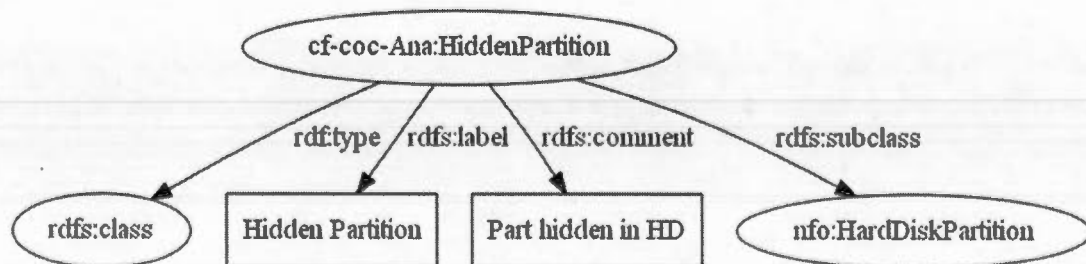


Figure 7.46 RDF model of the “*HiddenPartition*” class

Now, Robert will define the term “contains” with as domain “SecondaryDevice” and as range the “HiddenPartition” class (see Figure 7.47 and Figure 7.48).

Term Name : *	contains (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Dublin_Core (dc)
	Property Name	hasPart (http://purl.org/dc/terms/hasPart)
<input checked="" type="checkbox"/> Range	From	Custom Ontology
	Ontology Name	Analysis (cf-coc-Ana)
	Class Name	HiddenPartition (Analyze)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Authentication (cf-coc-Aut)
	Class Name	SecondaryDevice (Generate Checksum)
<input checked="" type="checkbox"/> Label	Enter a label for the term	contains

Figure 7.47 Screen for creating the “contains” property

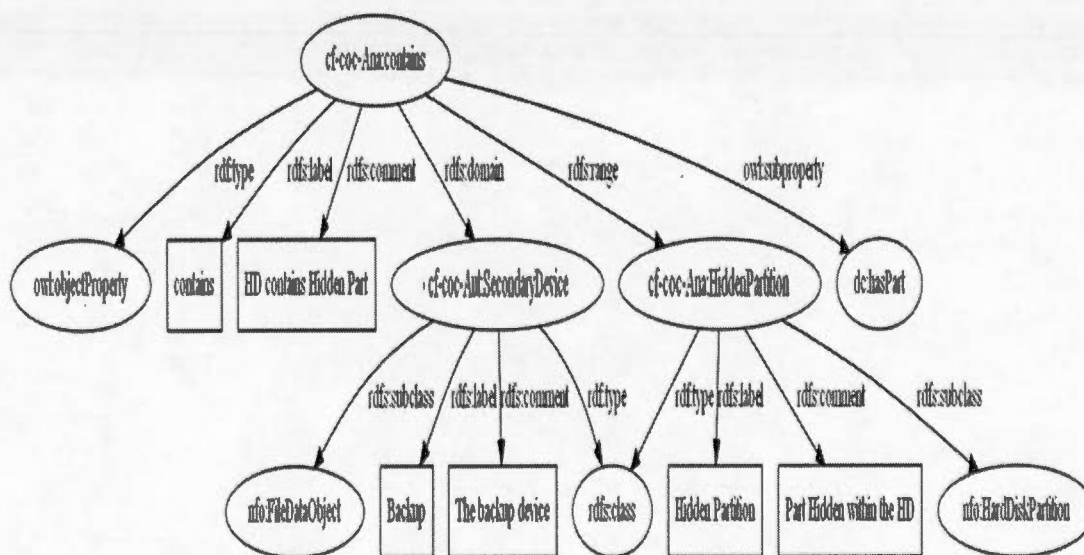


Figure 7.48 RDF model of the “contains” property

Attached with the “*HiddenPartition*” class, Robert creates also properties called “*hiddenContains*” and “*hiddenSize*”. For “*hiddenContains*”, it will be a sub-property of “*dc:hasPart*” (same idea illustrated by Figure 7.48): its domain will be the class “*HiddenPartition*”, and its range will be “*nie:InformationElement*”, which is a super-class for all interpretations data objects (i.e., super-class of all objects in the NIE ontology). The “*InformationElement*” class will refer to all excel files found in the hidden partition. For the size of hidden partition, it is same idea as Figure 7.44.

Term Name : *	hiddenContains (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Dublin_Core (dc)
	Property Name	hasPart (http://purl.org/dc/terms/hasPart)
<input checked="" type="checkbox"/> Range	From	Built-in Ontology
	Ontology Name	NEPOMUK_Information_Element (nie)
	Class Name	InformationElement (data objects)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Analysis (cf-coc-Ana)
	Class Name	HiddenPartition (Analyze)
<input checked="" type="checkbox"/> Label	Enter a label for the term hiddencontain	
<input checked="" type="checkbox"/> Comment	Enter a comment for the term hidden contents	

Figure 7.49 Screen for creating the “*hiddenContains*” property

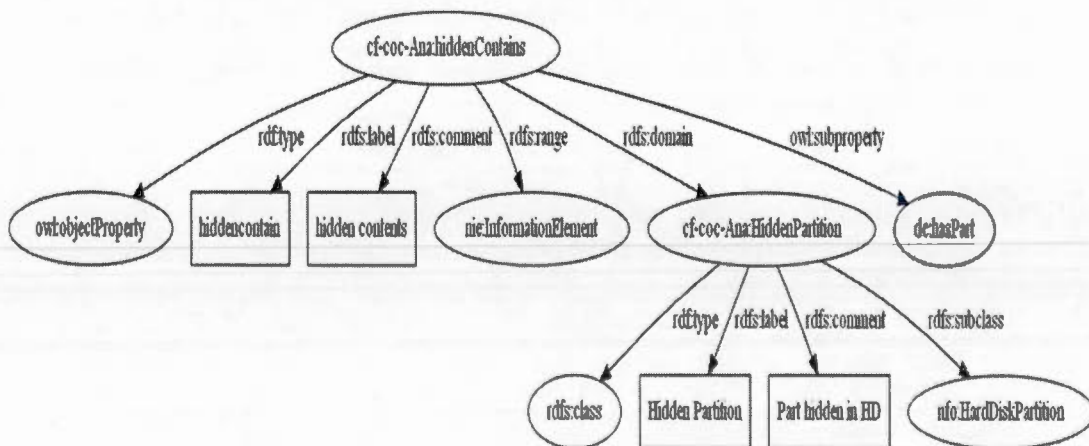


Figure 7.50 RDF model of the “*hiddenContains*” property

Robert also defines a property term, “*hiddenSize*”, as a sub-property of “*nie.contentSize*” (same idea as Figure 7.41 and 7.44). Its domain will be the hidden partition class “*HiddenPartition*” and its range will be “*xsd:integer*”.

The last property term Robert needs to define is the term defining the tool used by the fired employee to create a hidden part on the hard disk, where he stored the excel files. He calls this term “*hiddenUsing*”, and he defines it as a sub-property from the property “*foaf:made*”. Its domain is “*HiddenPartition*” and its range is the class “*Software*”, defined in the NFO ontology (see Figure 7.51 and Figure 7.52).

Term Name : *	hiddenUsing (Specify the name of the new term)	
In Ontology : *	Analysis (Specify in which ontology you define a new term)	
Category : *	<input type="radio"/> New <input checked="" type="radio"/> Existing	
Term Type : *	Property (Specify the type of the new term)	
RDF-Schema Vocabulary		
<input checked="" type="checkbox"/> Subproperty-of	From	Built-in Ontology
	Ontology Name	Friend_of_a_Friend (foaf)
	Property Name	made (Documents and Images)
<input checked="" type="checkbox"/> Range	From	Built-in Ontology
	Ontology Name	NEPOMUK_FILE_Ontology (nfo)
	Class Name	Software (Softwares)
<input checked="" type="checkbox"/> Domain	From	Custom Ontology
	Ontology Name	Analysis (cf-coc-Ana)
	Class Name	HiddenPartition (Analyze)
<input checked="" type="checkbox"/> Label	Enter a label for the term hidden using	

Figure 7.51 Screen for creating the “*hiddenUsing*” property

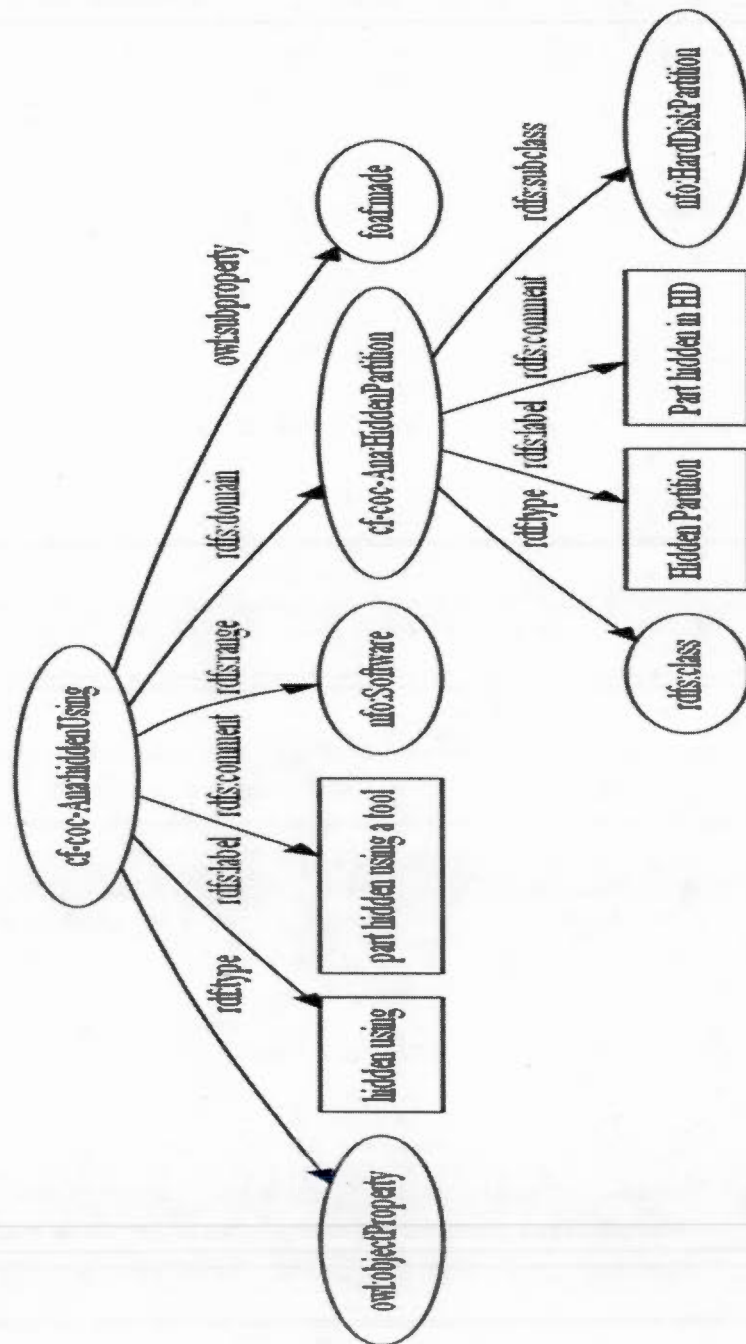


Figure 7.52 RDF model of the “*hiddenUsing*” property

7.3.3.2 The *e*-CoC of the analysis phase

After the definition of terms, Robert starts using the defined terms to publish and generate the *e*-CoC of his forensic phase. What he is doing during this analysis phase can be summarized into a set of triples:

1. Robert *analyze* Hard_drive_laptop
2. Hard_drive_laptop *dataSize* 100 Mega
3. Hard_drive_laptop *analyzedBy* Encase Tool
4. Hard_drive_laptop *totalSize* 105 Mega
5. Hard_drive_laptop *contains* hidden part
6. hidden part *hiddenContains* Excel Files
7. hidden part *hiddenSize* 5 Mega
8. hidden part *hiddenUsing* Secret Disk

Robert wishes also to integrate with his published *e*-CoC, the evidence format of the investigation analysis generated from the forensic tool. Assuming that the evidence is a file called “*evidence.aff4*”, and it is in the form of XML format describing different fields in the AFF4 namespace (see Figure 7.53). The corresponding RDF model of AFF4 is shown in Figure 7.55. Figure 7.54 shows the *e*-CoC generated by the CF-CoC, after publishing the above triples.

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:AFF4 xmlns:rdf=http://127.0.0.1/aff4#>
  <aff4:username>Rob</aff4>
  <aff4:MediaObject>
    <aff4:medianame>HardDisk</aff4:medianame>
    <aff4:size>5120</aff4:size>
  </aff4:Mediaobject>
</rdf:RDF>
```

Figure 7.53 RDF/XML Code for an AFF4 format

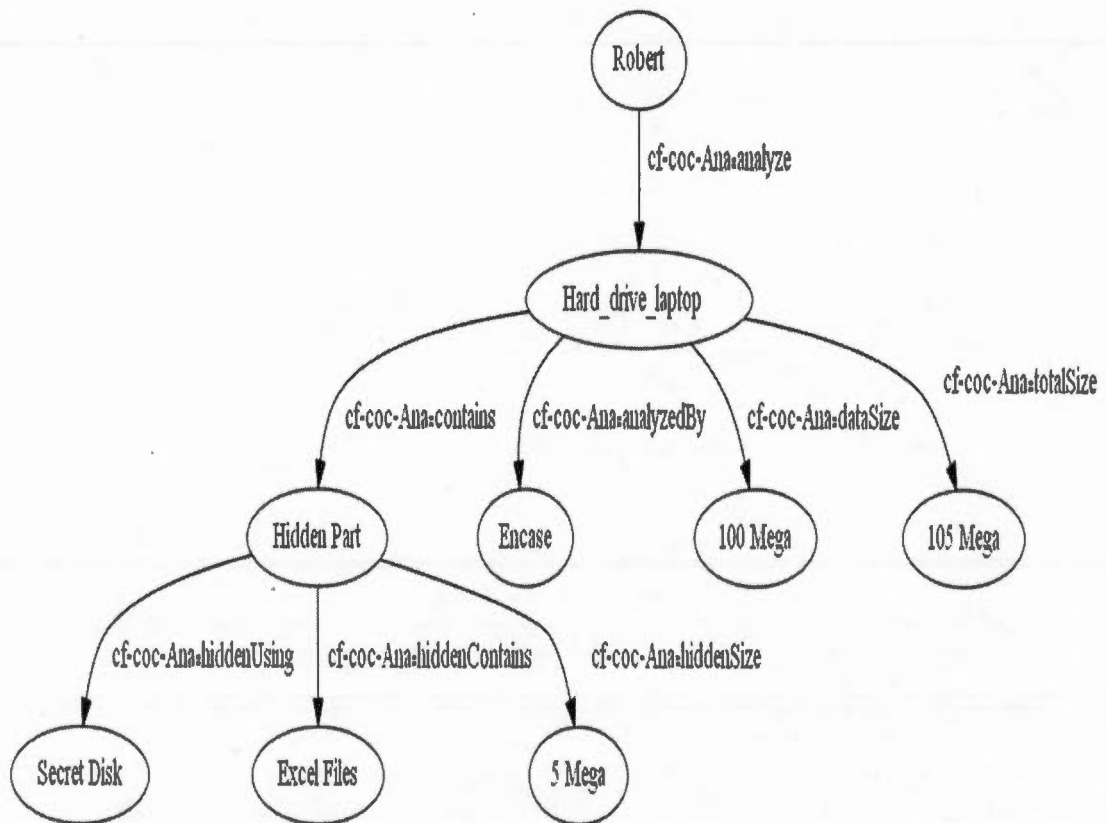


Figure 7.54 *e-CoC of the analysis phase*

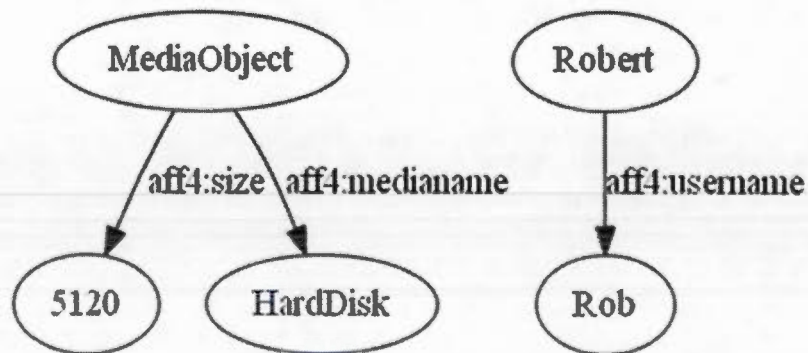


Figure 7.55 *RDF model for an AFF4 result*

Robert combines the results of the forensic tool with the *e*-CoC of the acquisition phase. The subject “*Robert*” is the same instance used in the analysis phase, so the “*aff4:username*” will be added as an extra property to the graph. The “*MediaObject*” is linked to the “*Hard_drive_laptop*” and its “*hiddenSize*”, using the “*sameAs*” property of OWL.

Finally, the three *e*-CoCs are combined together on the following figure (see Figure 7.56): the *e*-CoC of the acquisition, authentication, and analysis (i.e., including also the AFF4 information) are merged to get the complete *e*-CoC of the forensic case study (i.e., following the Kruse model provided in Figure 7.1).

As we see in Section 7.3, the CF-CoC system, which is based on representing information using LDP and creating RDF models, is able to produce *e*-CoCs readable and consumable by people and machines. The role players of each forensic phase use the system to represent and publish the forensic investigation results. A role player is able to create his own terms or import terms from other forensic phases created by other role players. This allows the creation of interlinked *e*-CoC between different phases, the fact that foster and improve the comprehension of the presented information.

Also, if a role player uses his own terms, he will be able, at a later stage, to combine what he published with other phases and getting an interlinked *e*-CoC by mapping his terms with other terms. In addition, a role player is able to attach results generated from forensic tools within the published *e*-CoC under the same RDF framework. This makes that the forensic information obtained by forensic tools interoperable with the information published by the role players.

7.4 Adding provenance information to *e*-CoCs

Section 7.3 discussed how the CF-CoC system is used to create and publish *e*-CoCs. These *e*-CoCs contain different forensic information related to the digital evidence in hand. Such information answers the five Ws and one H questions related to the forensic information. As explained in Chapter 2, there are three common elements that make evidence admitted to the court of law: authenticity, relevancy and reliability. Other information should be provided to explain the origin of this information, and answer the five Ws and one H questions about the origin of such information. This section will work to prove the second hypothesis that states the provenance information with the published *e*-CoCs can foster the trustworthiness among role players and judge. This will be useful for judge to know supplementary information about the forensic information published by the role players (e.g., when they did the publication, why and how they collected this information, what is the validity of such information, etc. These questions differ from one phase to another).

Generally, the provenance metadata are used to annotate the published information on the web then they are extracted from the web using various automated softwares. In this dissertation, the CF-CoC system is residing on owned domain somewhere on the web and uses the LDP to publish different resources. The system uses the digital certificates to restrict the access to these resources (see Section 7.2). The system can be used to annotate the provenance metadata but there is no need for automated softwares to extract them.

This section shows the annotation of *e*-CoCs by the three role players of this investigation case: Jean-Pierre, Peter, and Robert. They add supplementary information about themselves or about the data origin. The forensic information annotated in this section using the vocabularies of Dublin Core and Friend of a Friend

Function (FOAF) is based on the Named Graph (NG) approach. Other vocabularies can be used to annotate the forensic information.

7.4.1 Adding provenance metadata to the acquisition phase

Jean-Pierre in his annotation indicates that he is the main contributor of this phase, and his role is the first seizure in this investigation process. He also confirms that he creates the *e-CoC* complying with the tangible document itself. Also, the investigation is performed on the IT company (see Figure 7:57)

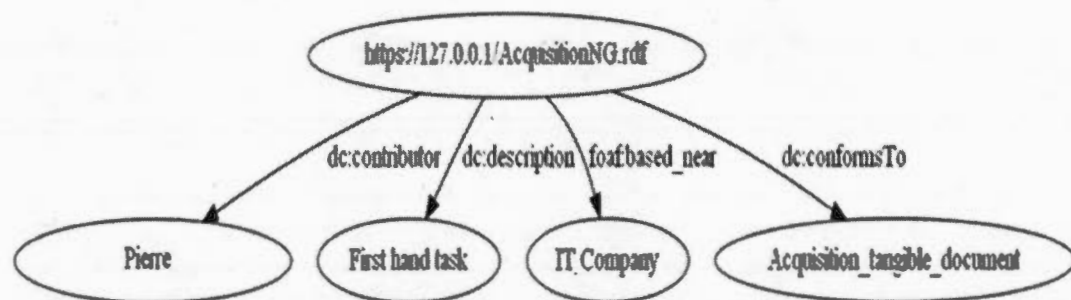


Figure 7.57 Provenance graph for the acquisition phase

7.4.2 Adding provenance metadata to the authentication phase

Assuming that Robert, the role player of the analysis phase, should be present during checking the integrity of both devices to confirm that the backup device is provided to Robert by hand, not by mail, and avoid the suspicion that the device has been altered along its route. Peter wanted to append this information using provenance metadata to confirm that after he finished his task, he provided the backup device by

hand to Robert for analyzing it. He uses a property called “*valid*”, imported from the Dublin Core vocabulary, to confirm that Robert was present when he did the integrity task.

He also mentions that the MD5 algorithm is required to accomplish this authentication phase. He mentions that the *e*-CoC for authentication is issued under his work license number, and the date he submitted the backup device to Robert is 1st of May 2015 (see Figure 7.58).

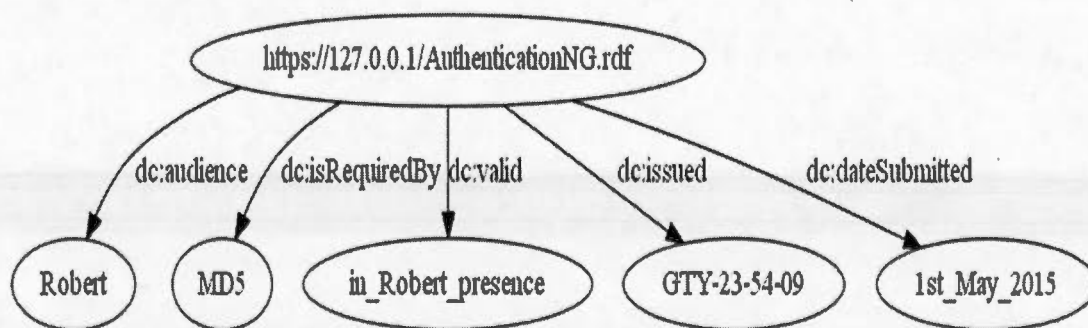


Figure 7.58 Provenance graph for the authentication phase

7.4.3 Adding provenance metadata to the analysis phase

Robert in his annotation mentions that he creates his *e*-CoC in 20th May 2015, he also states to refer to the tangible document related to the acquisition phase in order to compare it with the results of the analysis phase (i.e., in the acquisition phase Jean-Pierre only recovered word files, while in the analysis phase, Robert revealed that there were more hidden information on the device). In addition, Robert states that his *e*-CoC conforms to the tangible CoC of the analysis phase. He also provides the date of submission of the backup device from Peter, which is the same date he accepted it.

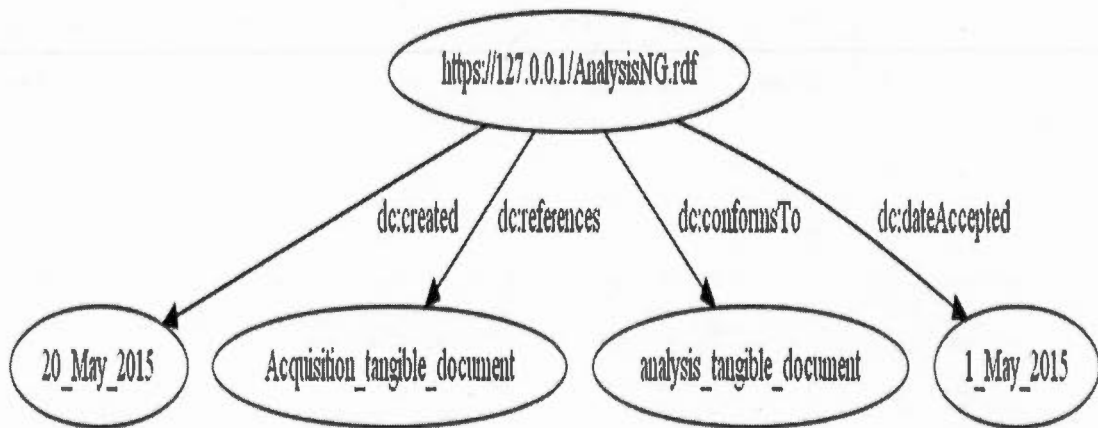


Figure 7.59 Provenance graph for the analysis phase

Other provenance metadata can be injected to each *e*-CoC to enrich the published forensic information. Also, custom property terms can be created to provide more information about them.

7.5 Applying the consumption patterns on the *e*-CoC of Kruse model case study

After role players publish their forensic information and annotate their *e*-CoCs using the provenance metadata, the judge can consume them using the consumption patterns module. As explained, the CF-CoC contains four consumption patterns. Such patterns help the judge to consume and understand the digital evidence and improve the subject matter, as it was proposed in the third hypothesis of this dissertation. The following patterns are not restricted to the judge only, but they can be used by any role player of the forensic process.

7.5.1 Browsing the *e*-CoC of Kruse model case study

Browsing allows the judge to display all phases together or each phase apart. If the judge chooses to display all phases together, he will be able to see the complete *e*-CoC of the Kruse model (see Figure 7.56), the provenance information associated to each forensic phase (see Figure 7.57, Figure 7.58, and Figure 7.59), and to see the ontology object of each forensic phase (i.e., general information about the forensic phase itself, its domain, publication date, and the role player certificate).

If the judge chooses to select each forensic phase apart, he will be able to navigate between different resources describing the forensic phase. This selection is used when the judge needs to navigate internally and get more details about the forensic tasks of each forensic phase (see Figure 7.60).

Forensic Information Consumption

Forensic Phase : *		Acquisition
		Search
Forensic Tasks For Acquisition Phase	Described by Resource(s)	
Preservation	https://127.0.0.1/Vocab/Acquisition#RolePlayer https://127.0.0.1/Vocab/Acquisition#FirstResponder https://127.0.0.1/Vocab/Acquisition#DigitalMedia https://127.0.0.1/Vocab/Acquisition#SN https://127.0.0.1/Vocab/Acquisition#preservedBy https://127.0.0.1/Vocab/Acquisition#preserve	
Recovery	https://127.0.0.1/Vocab/Acquisition#SuspectedDevice https://127.0.0.1/Vocab/Acquisition#recover https://127.0.0.1/Vocab/Acquisition#DeletedFiles https://127.0.0.1/Vocab/Acquisition#containsRecover	
Backup	https://127.0.0.1/Vocab/Acquisition#backup https://127.0.0.1/Vocab/Acquisition#BackupDevice	

Figure 7.60 Screen for displaying the acquisition phase resources

As we see from the above figure, the three tasks of the acquisition phase, with their corresponding resources, are displayed when the judge selects the acquisition phase. Let's say that the judge do not understand what is the "backup" resource. He simply

clicks on the backup device and get more information, then navigate among this resource (see Figure 7.61).

As shown in Figure 7.61, the screen of the backup task illustrates that this task is a sub-property of the property “made”. Its subject (domain) is a “FirstResponder” and its object (range) is the “SuspectedDevice”. If the judge cannot understand all this information, he reads the comments of the backup resource he gets that “*A backup task is a copy of information from a source to a destination device*”. He also dereferences the “made” property, he gets the definition of this term, it means: “*something that was made by this agent*” (Brickley et Miller, 2014). He continues to dereference the other resources on the same screen, he clicks on the “FirstResponder”, he gets that it is a “RolePlayer”, he clicks on it, he obtains that this resource is of type “person”, he clicks on the “person”, he gets that “*the person class represent people*” and so on. The judge is able anytime to expand/dereference any represented resources, in order to understand and get more information about them.

Figure 7.61 not only explains and defines each term, but also it displays all the instances published using these terms. For example, in Figure 7.61, it displays that “Jean-Pierre” was the “FirstResponder”, and he did a backup for the “PDA device”.

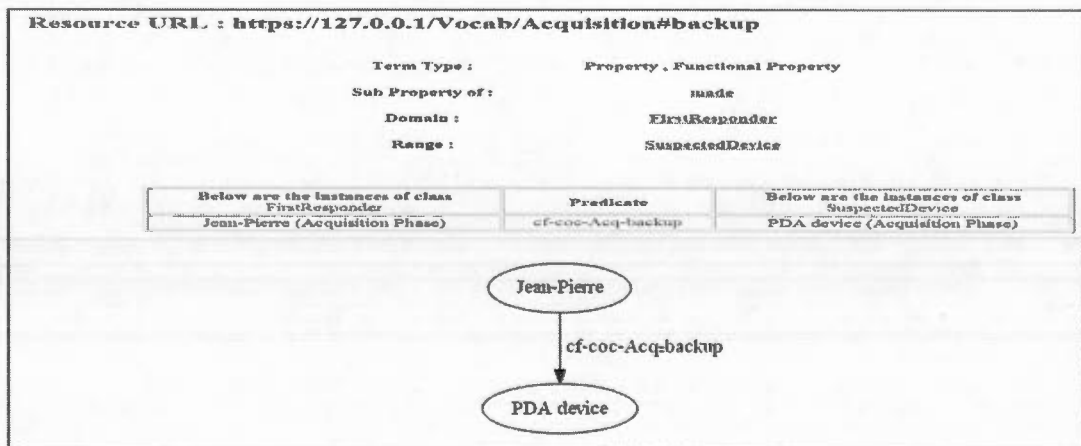


Figure 7.61 Screen of the backup task in the acquisition phase

The scenario will be the same for other tasks in other forensic phases. Users will be able to navigate between all resources defined within each forensic phase. Figure 7.62 and Figure 7.63, shows also the tasks of the authentication and analysis phases.

Forensic Information Consumption

Forensic Phase : *		Authentication
		Search
Generate Checksum		https://127.0.0.1/Vocab/Authentication#PrimaryDevice https://127.0.0.1/Vocab/Authentication#Authenticator https://127.0.0.1/Vocab/Authentication#authenticatePrimary https://127.0.0.1/Vocab/Authentication#ImagefilePrimary https://127.0.0.1/Vocab/Authentication#chckalgorithmPrimary https://127.0.0.1/Vocab/Authentication#SecondaryDevice https://127.0.0.1/Vocab/Authentication#authenticateSecondary https://127.0.0.1/Vocab/Authentication#ImagefileSecondary https://127.0.0.1/Vocab/Authentication#hashingSecondary https://127.0.0.1/Vocab/Authentication#hashingPrimary https://127.0.0.1/Vocab/Authentication#checksumSecondary https://127.0.0.1/Vocab/Authentication#checksumPrimary https://127.0.0.1/Vocab/Authentication#chckalgorithmSecondary

Figure 7.62 Screen for displaying the resources of the authentication phase

Forensic Information Consumption

Forensic Phase : *		Analysis
		Search
Forensic Tasks For Analysis Phase	Described by Resource(s)	
Analyze	https://127.0.0.1/Vocab/Analysis#Analyzer https://127.0.0.1/Vocab/Analysis#analyze https://127.0.0.1/Vocab/Analysis#dataSize https://127.0.0.1/Vocab/Analysis#ForensicTool https://127.0.0.1/Vocab/Analysis#analyzedBy https://127.0.0.1/Vocab/Analysis#totalSize https://127.0.0.1/Vocab/Analysis#HiddenPartition https://127.0.0.1/Vocab/Analysis#contains https://127.0.0.1/Vocab/Analysis#hiddenContains https://127.0.0.1/Vocab/Analysis#hiddenUsing https://127.0.0.1/Vocab/Analysis#hiddenSize	

Figure 7.63 Screen for displaying the resources of the analysis phase

7.5.2 Crawling the *e*-CoC of Kruse model case study

As explained before, the crawling is a search by a keyword. After publishing the three forensic phases, the judge can use the CF-CoC system to find and get information about any forensic resource in any forensic phase. A crawling using a keyword retrieves all triples containing this keyword, and appearing in any slot of any the RDF triples. Using this pattern, judge will be able to get any information from the *e*-CoC.

For instance, if the judge needs to get all tasks performed by Robert, he does his search using the keyword "*Robert*". The result will be two triples containing Robert as a subject as shown in Figure 7.64:

Enter a Literal Keyword for Crawling Triples: *

(Enter % to Crawl all Triples)

Triples		
Robert (Acquisition)	analyze (Analysis)	Hard_drive_laptop (Authentication)
Robert (Acquisition)	username (AFF4)	Rob (Analysis)

Figure 7.64 Screen for crawling using the keyword "*Robert*"

Another example is shown in Figure 7.65, by crawling using the keyword "*LaptopHD*".

Enter a Keyword for Crawling Triples: *
 (Enter % to Crawl all Triples)

Triples		
PDA device (Acquisition)	backupTo (Acquisition)	Laptop HD (Acquisition)
Laptop HD (Acquisition)	sameAs (Ontology_Web_Language)	Hard_drive_laptop (Authentication)

Figure 7.65 Screen for crawling using the keyword “*LaptopHD*”

7.5.3 Reasoning the *e*-CoC of Kruse model case study

To explain clearly the mechanism of the reasoning module, classes and properties for each forensic task are summarized in the following tables. This will facilitate how the system infers implicit information (i.e., triples that are not explicitly stored in the RDF store) using the entailment rules mentioned in Chapter 2. The inferred information is a set of resources that are also interlinked and dereferenced to more other resources (see Section 7.5.1). The inferred information is not limited to the examples mentioned in the next sections; more information can be inferred using other examples.

7.5.3.1 Reasoning on the acquisition phase

The classes of the acquisition phase are provided in the next table. This table provides all terms of type class that appear in domain and range of properties.

Table 7.2 Classes of the acquisition phase

Classes					
Phase	Task	Term name	In Domain of	In Range of	Subclass of
Acquisition	Preservation	RolePlayer	-	-	Person
		FirstResponder	backup, preserve, recover	preservedBy	RolePlayer
		DigitalMedia	SN, preservedBy	preserve	Things
	Backup	BackupDevice	-	backupTo	DigitalMedia
	Recovery	SuspectedDevice	backupTo, containsRecover	backup, recover	DigitalMedia
		DeletedFiles	-	contains_recover	DeletedResource

The properties of the acquisition phase are provided in the next table. It shows the domain, range, and sub-property of all property terms used by each forensic task.

Table 7.3 Properties of the acquisition phase

Properties					
Phase	Task	Term name	Domain	Range	Sub-property of
Acquisition	Preservation	SN	DigitalMedia	Literal	identifier
		preservedBy	DigitalMedia	First_Responder	made
		preserve	FirstResponder	Digital_Media	made
	Backup	backup	FirstResponder	Suspected_device	made
		backupTo	Suspected_Device	Backup_device	made
	Recovery	recover	First_Responder	Suspected_device	made
		containsRecover	Suspected_Device	Deleted_Files	format

Referring to the Table 2.3, next points provide some examples to illustrate how the reasoning module of the CF-CoC works with the classes and properties of the acquisition phase. For each RDFS constructor, an example is provided, to illustrate

how its entailment rule(s) can be applied using the terms schema of T-Box (i.e., classes and properties) and A-Box (i.e., instances).

- *rdfs:subClassOf*

For $(A, \textit{rdfs:subClassOf}, B), (B, \textit{rdfs:subClassOf}, C) \Rightarrow (A, \textit{rdfs:subClassOf}, C)$

$(\textit{FirstResponder}, \textit{rdfs:subClassOf}, \textit{RolePlayer}), (\textit{RolePlayer}, \textit{rdfs:subClassOf}, \textit{Person}) \Rightarrow (\textit{FirstResponder}, \textit{rdfs:subClassOf}, \textit{Person})$

Another Entailment rule of *rdfs:subClassOf*:

$(a, \textit{rdf:type}, A), (A, \textit{rdfs:subClassOf}, B) \Rightarrow (a, \textit{rdf:type}, B)$

$(\textit{Jean-Pierre}, \textit{rdf:type}, \textit{FirstResponder}), (\textit{FirstResponder}, \textit{rdfs:subClassOf}, \textit{RolePlayer}) \Rightarrow (\textit{Jean-Pierre}, \textit{rdf:type}, \textit{RolePlayer})$

- *rdfs:subPropertyOf*

For $(a, p, b) (p, \textit{rdfs:subPropertyOf}, q) \Rightarrow (a, q, b)$

$(\textit{Jean-Pierre}, \textit{recover}, \textit{PDA device}), (\textit{recover}, \textit{rdfs:subPropertyOf}, \textit{made}) \Rightarrow (\textit{Jean-Pierre}, \textit{made}, \textit{made}, \textit{PDA device})$

- *rdfs:domain*

$(p, \textit{rdfs:domain}, A), (a, p, x) \Rightarrow (a, \textit{rdf:type}, A)$

$(\textit{containsRecover}, \textit{rdfs:domain}, \textit{SuspectedDevice}), (\textit{PDA device}, \textit{containsRecover}, \textit{Word_files}) \Rightarrow (\textit{PDA device}, \textit{rdf:type}, \textit{SuspectedDevice})$

- *rdf:range*

$$(p, \text{rdfs:range}, A), (x, p, a) \Rightarrow (a, \text{rdf:type}, A)$$

(*preserve*, *rdfs:range*, *DigitalMedia*), (*Jean-Pierre*, *preserve*, *PDAdevice*) \Rightarrow (*PDA device*, *rdf:type*, *DigitalMedia*). The last two inferences deduced that the “*PDA device*” is of type “*SuspectedDevice*” and at the same time of type “*DigitalMedia*”. By referring to Table 7.2, the “*SuspectedDevice*” is a subclass of “*DigitalMedia*”. This asserts the same information that can be inferred from the following entailment rule:

$$(a, \text{rdf:type}, A), (A, \text{rdfs:subClassOf}, B) \Rightarrow (a, \text{rdf:type}, B)$$

$$(\text{PDA device}, \text{rdf:type}, \text{SuspectedDevice}), (\text{SuspectedDevice}, \text{rdfs:subClassOf}, \text{DigitalMedia}) \Rightarrow (\text{PDA device}, \text{rdf:type}, \text{DigitalMedia})$$

owl:FunctionalProperty, *owl:InverseFunctionalProperty* and *owl:inverseof*.

The complete examples of these three constructors in the acquisition phase are described in Section 5.4.

7.5.3.2 Reasoning on the authentication phase

The classes of the authentication phase are provided in the next table. This table provides all terms of type class that appear in domain and range of properties.

Table 7.4 Classes of the authentication phase

Classes					
Phase	Task	Term name	In Domain of	In Range of	Subclass of
Authentication	Generate Checksum	Primary_ Device	hashing_ Primary	authenticate_ primary	FileDataObject
		Secondary_ Device	hashing_ Secondary	authenticate_ secondary	FileDataObject
		Authenticator	authenticate_ Primary, authenticate_ Secondary	-	RolePlayer
		Imagefile_ Primary	checksum_ primary, chckalgorithm_ Primary	hashing_ primary	FileHash
		Imagefile_ Secondary	checksum_ Secondary, chckalgorithm_ Secondary	Hashing_ Secondary	FileHash

The properties of the authentication phase are provided in the next table. This table shows the domain, range, and sub-property of all properties term used by each forensic task (i.e., generate checksum) in the authentication phase.

Table 7.5 Properties of the authentication phase

Properties					
Phase	Task	Term name	Domain	Range	Sub-property of
Authentication	Generate Checksum	authenticate_ Primary	Authenticator	Primary_ Device	made
		authenticate_ Secondary	Authenticator	Secondary_ Device	made
		hashing_ Primary	Primary_ Device	Imagefile_ Primary	hasHash
		hashing_ Secondary	Secondary_ Device	Imagefile_ Secondary	hasHash
		checksum_ Primary	Imagefile_ Primary	String	hasValue
		checksum_ Secondary	Imagefile_ secondary	String	hasValue
		chckalgorithm_ Primary	Imagefile_ Primary	String	hasAlgorithm
		chckalgorithm_ Secondary	Imagefile_ Secondary	String	hasAlgorithm

Referring to the Table 2.3, next points provide some examples to illustrate how the reasoning module of the CF-CoC works with the classes and properties of the authentication phase.

For each RDFS constructor, an example is provided, to illustrate how its/their entailment rule(s) can be applied using the terms schema of T-Box (i.e., classes and properties) and A-Box (i.e., instances).

- *rdfs:subClassOf*

For $(A, rdfs:subClassOf, B), (B, rdfs:subClassOf, C) \Rightarrow (A, rdfs:subClassOf, C)$

$(ImagefileSecondary, rdfs:subClassOf, FileHash), (FileHash, rdfs:subClassOf, FileDataObject) \Rightarrow (ImagefileSecondary, rdfs:subClassOf, FileDataObject)$

Another Entailment rule of *rdfs:subClassOf*:

$(a, rdf:type, A), (A, rdfs:subClassOf, B) \Rightarrow (a, rdf:type, B)$

$(HDL_image.img, rdf:type, ImagefileSecondary), (ImagefileSecondary, rdfs:subClassOf, FileHash) \Rightarrow (HDL_image.img, rdf:type, FileHash)$

- *rdfs:subPropertyOf*

For $(a, p, b), (p, rdfs:subPropertyOf, q) \Rightarrow (a, q, b)$

$(Personal_Digital_Assistant, hashingPrimary, PDA_image.img), (hashingPrimary, rdfs:subPropertyOf, hasValue) \Rightarrow (Personal_Digital_Assistant, hasValue, PDA_image.img)$

- *rdfs:domain*

$$(p, \text{rdfs:domain}, A), (a, p, x) \Rightarrow (a, \text{rdf:type}, A)$$

$$(\text{checksumPrimary}, \text{rdfs:domain}, \text{ImagefilePrimary}), (\text{PDA_image.img}, \text{checksumPrimary}, \text{MD5}) \Rightarrow (\text{PDA_image.img}, \text{rdf:type}, \text{ImagefilePrimary})$$

- *rdfs:range*

$$(p, \text{rdfs:range}, A), (x, p, a) \Rightarrow (a, \text{rdf:type}, A)$$

$$(\text{hashingSecondary}, \text{rdfs:range}, \text{ImagefileSecondary}), (\text{Hard_drive_laptop}, \text{hashingSecondary}, \text{HDL_image.img}) \Rightarrow (\text{HDL_image.img}, \text{rdf:type}, \text{ImagefileSecondary})$$

- *owl:FunctionalProperty*

In our case study, there is one triple published to describe the hashing task of the laptop' hard drive:

hashingSecondary (hard_drive_laptop, HDL_image.img)

Let us assume that there was another triple describing the hashing task of the device using an instance called "*Laptop_Hard_image.img*"

hashingSecondary (hard_drive_laptop, Laptop_Hard_image.img)

If a property *p* is tagged as a *FunctionalProperty* then all *x*, *y*, and *z*: *p* (*x*, *y*) and *p* (*x*, *z*) $\Rightarrow y=z$

So if we have the following two triples and *hashingSecondary* is tagged as *FunctionalProperty*:

```

hashingSecondary (hard_drive_laptop, HDL_image.img) and
hashingSecondary (hard_drive_laptop, Laptop_Hard_image.img) =>
Hard_image.img = Laptop_Hard_image.img

```

Thus, when the judge consumes one of them, the system manipulates and provides all related resources of both “*HDL_image.img*” and “*Laptop_Hard_image.img*”. Both will be complementary information to each others. This means that both instances are the same, and considered as they are related together using the “*sameAs*” property.

- *owl:sameAs*

In the Figure 7.56, there was a mapping using the “*sameAs*” between “*LaptopHD*” (i.e., used by Jean-Pierre) in the acquisition phase and the “*Hard_drive_laptop*” (i.e., used by Peter in the authentication phase). As mentioned, the relation between these two instances made that the resources related to both instances are complementary information to each others.

The system provides to the judge that the “*LaptopHD*” is the backup of “*PDA device*”, and the “*LaptopHD*” is the same instance of “*Hard_drive_laptop*”, which has been hashed to generate the “*HDL_image.img*”.

- *owl:InverseFunctionalProperty*

In our case study, there is one triple published to describe the value of the checksum generating from primary image “*PDA_image.img*”:

```
checksumPrimary (PDA_image.img, 0X49E9DEC3)
```

Let us assume that there was another triple describing the checksum, and a new instance is used to describe the PDA image called: “IMG-Personal assistant”:

checksumPrimary (IMG-Personal assistant, 0X49E9DEC3)

If a property *p* is tagged as *InverseFunctionalProperty* then all *x*, *y* and *z*:
 $p(y,x) \text{ and } p(z,x) \Rightarrow y=z$

So, if we have the following two triples and “*checksumPrimary*” is tagged as *InverseFunctionalProperty*:

checksumPrimary (PDA_image.img, 0X49E9DEC3) and *checksumPrimary* (IMG-Personal assistant, 0X49E9DEC3) \Rightarrow PDA_image.img = IMG-Personal assistant.

7.5.3.3 Reasoning on the analysis phase

As same footstep of the two last sections, next table shows the classes of the analysis phase and provides the domain, range, and subclass of the forensic analyze task (see table 7.6). The “*SecondaryDevice*” is a class term defined by Peter in the generate checksum task of the authentication phase, and it is reused by Robert in the reasoning task of the analysis phase. It is mentioned in Table 7.6 because it is used as a domain and range of analysis phase properties.

Table 7.6 Classes of the analysis phase

Classes					
Phase	Task	Term name	In Domain of	In Range of	Subclass of
Analysis	Analyze	Analyzer	analyze	-	Role_ player
		Forensic_ tool	-	analyzedBy	Software
		Hidden_ partition	hidden_ contains, hidden_ using, hidden_ size	contains	HardDiskPartition
Imported from the Authentication phase	Generate Checksum	Secondary Device	analyzedBy, dataSize, totalSize, contains	analyze	FileDataObject

The properties of the analysis phase are provided in the next table. This table shows the domain, range, and sub-property of all property terms in the analysis phase.

Table 7.7 Properties of the analysis phase

Properties					
Phase	Task	Term name	Domain	Range	Sub-property of
Analysis	Analyze	analyze	Analyzer	Secondary_device	made
		analyzedBy	Secondary_Device	ForensicTool	made
		dataSize	Secondary_Device	Integer	contentSize
		totalSize	Secondary_Device	Integer	contentSize
		contains	Secondary_Device	HiddenPartition	hasPart
		hidden_Contains	Hidden_Partition	InformationElement	hasPart
		hidden_Using	Hidden_Partition	Software	made
		hiddenSize	Hidden_Partition	Integer	contentSize

- *rdfs:subClassOf*

For $(A, rdfs:subClassOf, B), (B, rdfs:subClassOf, C) \Rightarrow (A, rdfs:subClassOf, C)$

$(Analyzer, rdfs:subClassOf, RolePlayer), (RolePlayer, rdfs:subClassOf, Person) \Rightarrow (Analyzer, rdfs:subClassOf, Person)$

Another Entailment rule of *rdfs:subClassOf*:

$(a, rdf:type, A), (A, rdfs:subClassOf, B) \Rightarrow (a, rdf:type, B)$

$(Encase, rdf:type, ForensicTool), (ForensicTool, rdfs:subClassOf, Software) \Rightarrow (Encase, rdf:type, Software)$

- *rdfs:subPropertyOf*

For $(a, p, b), (p, rdfs:subPropertyOf, q) \Rightarrow (a, q, b)$

$(Hard_drive_laptop, contains, hidden_part), (contains, rdfs:subPropertyOf, hasPart) \Rightarrow (Hard_drive_laptop, hasPart, hidden_part)$

- *rdfs:domain*

$(p, rdfs:domain, A), (a, p, x) \Rightarrow (a, rdf:type, A)$

$(hiddenContains, rdfs:domain, HiddenPartition), (hidden_part, hiddenContains, Excel\ Files) \Rightarrow (hiddenContains, rdf:type, HiddenPartition)$

- *rdfs:range*

$(p, rdfs:range, A), (x, p, a) \Rightarrow (a, rdf:type, A)$

(analyzedBy, *rdfs:range*, ForensicTool), (Hard_drive_laptop, analyzedBy, Encase) => (Encase, *rdf:type*, ForensicTool)

- *owl:FunctionalProperty*

If a property *p* is tagged as a “*FunctionalProperty*” then all *x*, *y*, and *z*: *p* (*x*, *y*) and *p* (*x*, *z*) => *y*=*z*

The “*analyze*” property in the analysis phase is tagged as “*FuntionalProperty*”, because it is assumed that the role player Robert of the analysis phase, wants to assert that there is only one media to investigate during his phase, and this media is the “*Hard_drive_laptop*”.

The triple that describe the analyze task is (Robert, *analyze*, Hard_drive_laptop). If there exist another resource describing the laptop hard drive, for example, “*HDL*” that stands for hard drive laptop, in a triple (Robert, *analyze*, HDL) => Hard_drive_laptop = HDL, and both are referring to the same concept.

- *owl:InverseFunctionalProperty*

there is no property term defined by “*Robert*” in the analysis phase that is tagged as an “*InverseFunctionalProperty*”.

7.5.4 Querying the *e*-CoC of Kruse model case study

Using this pattern, a judge can query the three phases of the Kruse model. This module has an interface to use the SPARQL code and query the triples from the RDF store. The main difference between this pattern and the crawling one is that, in this pattern, judge is not aware about the published information. He can exploit this

pattern for first time consumption in order to retrieve and discover as much as he can from the published information (i.e., all instances are listed in each slot within the triple). However, in the crawling pattern, judge knows some information about the published resources, and he can crawl using one of these resources. Figure 7.66 shows an example of the querying pattern.

Forensic Information Consumption

Subject	Predicate	Object
Hard_drive_laptop ▾	- All Predicate (default) - ▾	- All Objects (default) - ▾
Query Now		
Hard_drive_laptop	hashingSecondary	HDL_image.img
Hard_drive_laptop	contains	Hidden Part
Hard_drive_laptop	analyzedBy	Encase
Hard_drive_laptop	dataSize	100 Mega
Hard_drive_laptop	totalSize	105 Mega

Figure 7.66 Screen for querying using “*Hard_drive_laptop*”

This interface avoids the judge to write down the SPARQL code to query the RDF forensic triples.

7.6 Conclusion

This chapter discussed how the CF-CoC system can be applied on a complete forensic process. The implemented module in this chapter answered and proved the research hypotheses.

It started with Section 7.2, the PKI module (i.e., sixth module in the framework). It depicts how the digital certificates are issued and used by the role players and neutral side, to consume the represented resources of LD on a closed scale, and this part proved the fourth hypothesis:

“PKI can be applied to the Linked Data (LD) to securely publish and consume the data between role players and judges, as well as transform the open data to closed data”.

In Section 7.3, after role players are identified through their client certificates, they started to use the system to represent and convert their tangible CoCs into electronic data, and this is achieved through the modules of defining and publishing resources, and RDF triples (i.e., first, second, and third module in the framework). These modules were used to build interlinked *e-CoCs* using RDF models that can be used by people and machines and integrate the AFF4 results (see Section 7.3.3.2) within the published *e-CoCs*. This section proved the first hypothesis:

“The semantic web can be a fertile land to create interlinked e-CoCs, which are readable and consumable by people and machines, and the forensic information resulting from a forensic tool can be interoperable with these interlinked CoCs”.

After creating the trustworthiness among role players and judge using the PKI module, Section 7.4 explained how the role players started to annotate their forensic information using different provenance vocabularies imported from the semantic web (i.e., the fourth module in the framework). This module depicted how the role players use such metadata to foster and add supplementary provenance information about the origin of the published resources. This proved the second hypothesis:

“Provenance metadata of the semantic web can be useful to answer the questions about the origin of the CoC data, and then foster trustworthiness among role players and judges”.

Finally, Section 7.5 discussed how the judge or any role player used different consumption patterns of the LD to consume the forensic resources, in order to understand different forensic resources and consume all their instances to take the proper decision (i.e., fifth module in the framework). Those consumption patterns

helped the judge to dereference, query, browse, crawl, and infer the forensic resources, and foster the subject matter. This proved the third hypothesis:

“Representing the CoC resources using the linked data principles can provide a descriptive e-CoC and then improve the subject matter and the understanding of the digital evidence”.

CHAPTER VIII

CONCLUSION AND FUTURE WORK

8.1 Introduction

This dissertation discussed the era of using the Linked Data Principles (LDP) to represent information in a linked data manner for the web of data. It depicted how such principles are used together with the semantic web vocabularies to represent different resources. This has been elaborated through on an example imported from the forensic domain to represent a Chain of Custody (CoC) generated from each forensic phase. Other situations can be imported from other domains, such as insurance, health care, government, law enforcement, etc. By applying the LDP to the forensic domain, a new framework solution related to the field of *e-Justice* has been provided.

Generally, this dissertation depicted a novel framework that will be used by role players to represent tangible chains of custody resulting from their cyber investigation. Role players use this framework to represent and publish forensic resources in order to be consumed by judges in a court of law. This work explained in detail all layers of the framework that are based on the technology stack of the linked data. This technology stack (RDF, HTTP, and URI) is used to represent and publish different resources in a structured way on the web. The role players start their representation process by defining new proprietary/custom terms describing the forensic information of their tangible chain of custody. This task is performed using lightweight ontologies through RDFS constructors and some primitives from OWL.

Role players may also add different provenance metadata imported from semantic web vocabularies to describe the origin of forensic information and then strengthen the trustworthiness with judges. All represented resources are then published in RDF format upon URI resolution in order to be shared on a closed scale between role players and judges, through the public-key infrastructure approach. The latter is a research called the Linked Closed Data (LCD), the counterpart to Linked Open Data. Linked Closed Data share all the advantages of LOD, but come with consumption restrictions.

8.2 Organization and scopes

This dissertation was organized according to four main topics. These topics are manipulated in terms of challenges, objectives, proposed hypotheses, related research, and lastly, used methods and approaches. Finally, the hypotheses are proven by applying our novel framework to a complete forensic process.

The first topic dealt with the benefits of transforming the tangible CoC, describing the digital investigation to accommodate digital technologies. These benefits arise because such documents containing the information about digital evidence need also to be interoperable with the information generated from forensic investigation tools. Therefore, the proposal was that the tangible documents need to undergo a radical transformation from paper documents to electronic data, readable by humans and consumable by computers.

The dissertation proposed that the semantic web may provide fruitful insight for meeting this objective. This is proven first by illustrating the aspects used to create linked data on the web, how the LDP are used to create open linked data using lightweight ontologies (i.e., RDFS and some constructors from OWL), how

publishers can define their own proprietary terms using RDFS and how reasoning can be performed over the published resources. Secondly, it discussed all published works from the literature related to forensic representation and formats.

After that, the research methodology chapter states different advantages of using the LDP to transform the tangible CoC to *e*-CoC.

The second topic concerned fostering trustworthiness among role players and judges. This topic dealt with the fact that providing forensic resources should be accompanied and annotated by supplementary information describing the origin of this information. Therefore, the objective was to foster trust among judges and role players.

The dissertation proposed that the provenance metadata of the semantic web can be useful to answer the questions relating to the origin of the CoC data. This is covered by describing the different type of provenance vocabularies on the semantic web. Then, in the research methodology, the Named Graph approach was used to annotate the forensic information with provenance metadata.

The third topic tackled judges' awareness of digital evidence. This is a concern as judges do not usually have enough knowledge about the field of ICT. Therefore, the objective is to provide consumption patterns aiding judges to consume and understand the represented resources and foster the subject matter.

The dissertation proposed that representing resources using LDP can provide a descriptive and deferenceable *e*-CoC. This is covered by describing all types of LD consumption patterns. Subsequently, in the research methodology, the dissertation discussed the fact that these patterns can be alleviated in a way to separate judges from technical details.

The last topic dealt with adapting the PKI approach to the LOD. We argue that the openness of cyber forensics data and metadata would not be convenient. Some access restrictions should be applied. The objective was to share forensic information on a closed scale and bend the LOD into LCD.

This dissertation proposed that the PKI can be applied to publish and consume the data among role players and judge securely. This point is covered by first introducing the PKI and digital certificates, and then the purposes, protocols and types of digital certificates. In the research methodology, the dissertation then discussed the feasibility of applying the digital certificates to the LD.

Finally, in Chapter 7, the dissertation provided a complete case study applied to the Kruse model to validate the CF-CoC system and prove the proposed hypothesis. In this chapter, we explained how the modules in the novel framework answered all research problems, as well as how the system can be used to represent, consume, and secure forensic resources of a complete forensic task.

8.3 Contribution to the computer science dimension

The proposed framework contributed to the computer science dimension with the following points:

- Transforming tangible documents of a forensic process into electronic resources using linked data principles.
- Using provenance vocabularies of the semantic web to annotate the forensic resources.
- Adapting consumption patterns of the linked data to be used by the actors of a forensic process.

- Adapting the PKI approach to transform open linked data into closed linked data.
- Representing the forensic information in the form of an RDF model, allowing for the interoperability with the AFF4 format.

8.4 Contribution to the cognitive dimension

The proposed framework contributed to the cognitive dimension with the following points:

- Using the web aspects of the semantic web for formal representation of forensic information: state of the art did not mention that the tangible CoC has been represented before using the LDP. This is the first work to represent the tangible CoC using web aspects. By transforming the tangible CoC into *e*-CoC, we have a new formal representation of forensic information.
- Improving the subject matter for the judge in a court of law: by using the consumption patterns of the LD, the judge was able to consume and understand the represented resources by dereferencing, querying, crawling and reasoning. The LDP provided said patterns.
- This linked data structure allows role players to contribute as a collective in publishing their forensic information: this structure allows the role player to participate together in co-operation to construct a complete *e*-CoC for a forensic process.
- Representation of conceptual models for forensic processes and tasks using UML.

8.5 Limitations and future work

In terms of limitations, while we have tested our framework on the Kruse model, we aim to test it on other forensic models in the future. We are also in the process of collaborating with the cyber justice lab of the *Université de Montréal* (UDEM).³⁸ The lab's main objective is to integrate the field of ICT within the judicial system. Our solution framework helps to achieve that goal, as it is an example of using technology to serve the judicial system.

On the other hand, the implemented CF-CoC system used different technologies from different disciplines. The system is not limited to such technologies. However, it can be enhanced and upgraded according to the recent technologies provided in each discipline. The following sub-sections discuss different perspectives for said upgrades.

The ability of the system to infer more implicit rules can be enriched by the injection of more entailment rule set provided by OWL DL and OWL Full. These rules will help the consumers of the system to understand, publish and take the proper decisions about the crime case.

8.5.1 Framework modules

This framework assumed that the role players (i.e., publishers) already have enough capabilities to understand the web semantic technologies. However, this will not be always the case. We cannot expect the user to have the technical skills to understand such technologies (ontology, RDF, SPARQL, etc.), and therefore, be able to publish

³⁸ <http://www.cyberjustice.ca/>

the CoC data. It is necessary then to present an “intelligent” module that can guide the role players through the publication process. Following the model of an intelligent tutoring system, we can imagine having sub-modules within this module: one for the tutor that possesses the knowledge about teaching and tactics; one for the domain that possesses the knowledge about lightweight ontology; and one for students (i.e., role players) that reflects how much the student knows about the domain (Salgueiro et al., 2005).

This “intelligent” module will not only help the producers to publish their forensic resources. It will also guide the consumer (i.e., judge) to consume said resources through different online tutoring strategies (Frasson et al., 1996; Murray, 1999; Wenger, 2014). The judge may also misunderstand the contents of certain vocabularies in the court of law. An intelligent module based on linked education may help the judge to confirm and improve his comprehension about any forensic resources.

8.5.2 Semantic vocabularies

The semantic web is rich with different well-defined vocabularies. The current system does not encompass all these vocabularies. Said vocabularies can be added to support the creation of new proprietary terms and can also be added to support the level of provenance vocabularies (i.e., all provenance vocabularies can be found in (Hartig et Zhao, 2012)).

8.5.3 Creating ontologies for cyber forensics

Creating a common set of ontologies for cyber forensics will be very useful to publish different forensic resources using the lightweight ontology. This set will contain different terms to represent any information presented in the tangible CoC documents. Relating most closely to this section is the work mentioned in Section 2.2.1.5 (Brinson et al., 2006).

8.5.4 Machine consumption

Computer machines can be used to consume and learn from the represented information. This should be consumed through serializing down the RDF model. The usage of the machine in this framework is limited to extracting implicit information from explicit RDF triples stored in the RDF store, using some stored entailment-based rules. The system currently implemented is able to serialize down any RDF model to any serialized language (e.g., Turtle, RDF/XML, N3, etc.). In the future, this option allows for the exploitation of serialized code to extract common or uncommon patterns.

8.5.5 Linked Education

The area of Linked Education is booming nowadays. For educational purposes, it is provided according to the standard of the web of data and linked data. This new area has evolved thanks to the RDF standard for sharing data on the web. The CF-CoC is already based on this standard, which will integrate different educational resources and promote the interoperability between CF-CoC and said resources³⁹.

³⁹ <https://linkededucation.wordpress.com/>

BIBLIOGRAPHY

- Adida, B., Birbeck, M., McCarron, S. et Herman, I. (2004). RDFa Core 1.1. Retrieved April 2012 from <http://www.w3.org/TR/rdfa-syntax>.
- Al-Fedaghi, S. et Al-Babtain, B. (2012). Modeling the forensics process. *International journal of security and its application*, 6(4), 97-108.
- Alajbegović, A. H., Jamak, H. et Zečić, D. (2006). Digital signature algorithm. *International research/expert conference*, 665-668.
- Alexander, K., Cyganiak, R., Hausenblas, M. et Zhao, J. (2009). Describing Linked Datasets. *Linked data on the web workshops*.
- Alexander, K. et Hausenblas M. (2009). Describing linked datasets on the design and usage of void, the 'vocabulary of interlinked datasets. *Linked data on the web workshops*.
- Andrew, M. W. (2007). Defining a process model for forensic analysis of digital devices and storage media. *Systematic approaches to digital forensic engineering, IEEE international workshop*, 8, 16-30.
- Ballou, S. (2010). *Electronic crime scene investigation: a guide for first responders* (2nd edition). Washington: Diane Publishing Co.
- Barker, E., Burr, W., Jones, A., Polk, T., Rose, S., Smid, M. et Dang, Q. (2009). Recommendation for key management. Part 3: Application-specific key management guidance. NIST special publication 800-57.
- Beckett, D. (2014). RDF 1.1 N-Triples. Retrieved March 2014 from <http://www.w3.org/TR/n-triples/>.
- Beckett, D. et Berners-Lee, T. (2011). Turtle-RDF triple language. Retrieved October 2012 from <http://www.w3.org/TeamSubmission/turtle/>.
- Beckett, D. et McBride B. (2004). RDF/XML syntax specification (revised). W3C recommendation.

- Bellovin, S. M. et Merritt M. (1990). Limitations of the Kerberos authentication system. *ACM SIGCOMM Computer communication review*, 20(5), 119-132.
- Berners-Lee, T. (2006). Design issues: linked data. Retrieved April 2011 from <http://www.w3.org/DesignIssues/LinkedData.html>.
- Berners-Lee, T. et Connolly D. (2011). Notation3 (N3): a readable RDF syntax. Retrieved November 2011 from <http://www.w3.org/TeamSubmission/n3/>.
- Berners-Lee, T., Fielding, R. et Masinter, L. (2014). Uniform resource identifier (URI): generic syntax. Retrieved December 2012 from <http://www.ietf.org/rfc/rfc3986>.
- Berners-Lee, T., Hendler, J. et Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 28-37.
- Bernstein, D. E. (2001). Frye, Again: the past, present, and future of the general acceptance test. *Jurimetrics*, 41(3), 385-407.
- Berrueta, D., Phipps, Miles, J., A., Baker, T. et Swick, R. (2008). Best practice recipes for publishing RDF vocabularies. Working draft, W3C.
- Berry, D., Buneman, P., Wilde, M. et Ioannidis Y. (2003). Data provenance and annotation. National e-science workshop centre.
- Bizer, C., Heath, T. et Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services. Interoperability and Web applications: emerging concepts*, 205-227.
- Blaze, M., Feigenbaum., J. et Keromytis, A. (1999). Key note: trust management for public-key infrastructures. *Security Protocols*, 59-63.
- Bogen, A. C. et Dampier, D. A. (2004). Knowledge discovery and experience modeling in computer forensics media analysis. *International symposium on information and communication technologies*, 140-145.
- Bonatti, P. A., Hoganb, A., Polleresb, A. et Sauroa, L. (2011). Robust and scalable linked data reasoning incorporating provenance and trust annotations. *Web semantics: science, services and agents on the World Wide Web*, 9(2), 165-201.

- Bray, T., Paoli, J., Sperberg, C. M., Maler, Eve. et Yergeau, F. (2008). Extensible Markup Language (XML) 1.0 (5th edition). Retrieved January 2011 from <http://www.w3.org/TR/REC-xml/>.
- Brickley, D. et Miller, L. (2014). FOAF Vocabulary Specification 0.99. Retrieved May 2011, from <http://xmlns.com/foaf/spec/>.
- Brinson, A. et al. (2006). A cyber forensics ontology: creating a new approach to studying cyber forensics. *Digital Investigation*, 3, 37-43.
- Brown, C. L. (2009). *Computer Evidence: Collection and Preservation*. Laxmi Publications.
- Cabral, J. E., Chavan, A., Clarke, T. M. et Greacen, J. (2012). Using technology to enhance access to justice. *Harvard journal of law and technology*, 26, 241-269.
- Cameron, G. (2003). Provenance and pragmatics. Workshop on data provenance and annotation.
- Campbell, L. et MacNeill, S. (2010). The Semantic Web, Linked and Open Data: a briefing paper, JISC Center of education technology, interoperability and standards.
- Carrier, B. (2003). Defining digital forensic examination and analysis tools using abstraction layers. *International Journal of Digital Evidence*, 1(4), 1-12.
- Carrier, B. D. (2006). A hypothesis-based approach to digital forensic investigations. Presented in partial fulfillment of the requirements for the degree of philosophy in the Prudue University.
- Carroll, J., Bizer, C., Hayes, P. et Stickler, P. (2005). Named graphs. *Web semantics: science, services and agents on the semantic web*, 3(4), 247-267.
- Casey, E. (2014). *Digital Evidence and Computer Crime* (3rd edition). Waltham: Academic Press.
- Common Digital Storage Format (CDEF) working group. (2009). Digital forensic forensics research conference. Retrieved October 2012 from <http://www.dfrws.org/CDESF/>.

- Chan, P. K., Fan, W., Prodromidis, A. et Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. *Intelligent systems and their applications*, IEEE, 14(6), 67-74.
- Cheng, G. et Y. Qu (2009). Searching linked objects with falcons: approach, implementation and evaluation. *International journal on semantic web and information systems*, 5(3), 49-50.
- Ciardhuáin, S. Ó. (2004). An extended model of cybercrime investigations. *International journal of digital evidence*, 3(1), 1-22.
- Cobden, M., Jennifer, B., Nicholas, G., Les C. et Nigel S. (2011). A research agenda for linked closed data. *International workshop on consuming linked data*.
- Cohen, M., Garfinkel, S. et Schatz, B. (2009). Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *Digital Investigation*, 6, 857-868.
- Coppersmith, D. (1994). The data encryption standard and its strength against attacks. *IBM journal of research and development* 38(3): 243-250.
- Corby, O., Gaignard, A., Faron Zucker, C. et Montagnat, J. (2012). Kgram versatile inference and query engine for the web of linked data. *International joint conferences on web intelligence and intelligent agent technology*, 1, 121-128.
- Cosic, J. et Baca, M. (2010a). (Im)proving chain of custody and digital evidence integrity with time stamp. *International Convention MIPRO*, 1226-1230.
- Cosic, J. et Baca, M. (2010b). A framework to (im)prove chain of custody in digital investigation process. *Central European conference on information and intelligent systems*, 435-438.
- Common Vulnerabilities and Exposures (CVE) (2014). The Standard for Information Security Vulnerability Names: Heartbleed bug. Retrieved May 2014 from <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0160>.
- Davies, J. (2011). *Implementing SSL/TLS using cryptography and PKI*. Indianapolis: Wiley Publishing Inc.

- Shrobe, H. et Szolovits, P. (1993). What is a knowledge representation? Artificial intelligence magazine, 14(1), 17-33.
- Dublin Core Metadata Initiative (DCMI) (2015). Metadata innovation. Retrieved June 2015 from <http://dublincore.org/>.
- Dean, M., Schreiber, G. et Bechhofer, S. (2004). OWL web ontology language reference. W3C recommendation.
- Dietze, S. et al. (2013). Interlinking educational resources and the web of data: a survey of challenges and approaches. Program, 47(1), 60-91.
- Dietze, S., Yu, H. Q., Giordano, D., Kaldoudi, E., Dovrolis, N. et Taibi, D. (2012). Linked Education: interlinking educational resources and the web of data. Annual ACM symposium on applied computing, 366-371.
- Ding, L. et al. (2004). Swoogle: a search and metadata engine for the semantic web. ACM international conference on information and knowledge management, 652-659.
- Dunkel, J., Bruns, R. et Ossowski, S. (2006). Semantic e-learning agents. Enterprise information systems, 6, 237-244.
- Eckert, K. (2013). Provenance and annotations for linked data. International conference on Dublin core and metadata applications, 9-18.
- Entrust (2010). Public-key Infrastructure. Retrieved June 2011 from www.entrust.com/what-is-pki/.
- Evangelia, M. et al. (2011). Connecting medical educational resources to the linked data cloud: the mEducator RDF schema, store and API. International workshop on e-learning approaches for the linked data Age.
- Farouk, M. et Ishizuka, M. (2012). An inference based query engine for RDF data. Information retrieval and knowledge management, 331-334.
- Farrell, M. G. (1993). Daubert v. Merrell Dow Pharmaceuticals. Incorporation. Epistemology and legal process Cardozo L. Rev, 15, 2183-2197.

- Fielding, R., Lafon, Y. et Reschke, J. (2014). Hypertext Transfer Protocol (HTTP/1.1): Range Requests, RFC 7233.
- Finklea, K. M. et Theohary C. A. (2012). Cybercrime: conceptual issues for congress and US law enforcement. Congressional research service report.
- Freire, J., Koop, D. et Moreau, L. (2008). The Open Provenance Model: An Overview. International Provenance and Annotation Workshop Series (IPAW), 5272. 323-326.
- Freitas, A., Curry, E., Oliveira, J. G. et O'Riain, S. (2012). Querying heterogeneous datasets on the Linked Data: challenges, approaches, and trends. Internet computing, IEEE, 6(1), 24-33.
- Simson, G., David, M., Karl-Alexander, D., Christopher, S. et P., Cecile (2006). Disk imaging with the advanced forensic format, library and tools. Research advances in digital forensics, 1-19.
- Gayed, T. F., Lounis, H. et Bari, M. (2012a). Computer forensics: toward the construction of electronic chain of custody on the semantic web. International conference on software engineering and knowledge engineering, 406-411.
- Gayed, T. F., Lounis, H. et Bari, M. (2012b). Cyber forensics: representing and (im) proving the chain of custody using the semantic web. International conference on advanced cognitive technologies and applications, 19-23.
- Gayed, T. F., Lounis, H. et Bari, M. (2013a). Representing chains of custody along a forensic process: a case study on Kruse model. International conference on software engineering and knowledge engineering, 674-680.
- Gayed, T. F., Lounis, H. et Bari, M. (2013b). Cyber forensics: representing and managing tangible chain of custody using the linked data principles. International conference on advanced cognitive technologies and application, 87-96.
- Gayed, T. F., Lounis, H. et Bari, M. (2014a). Linked closed data using PKI: a case study on publishing and consuming data in a forensic process. International conference on advanced cognitive technologies and applications, 77-86.

- Gayed, T. F., Lounis, H. et Bari, M. (2014b). Creating proprietary terms using lightweight ontology: a case study on acquisition phase in a cyber forensics process. *International conference on software engineering and knowledge engineering*, 76-8.
- Gayed, T. F., Lounis, H. et Bari, M. (2015). Representing and Publishing Cyber Forensic Data and its Provenance Metadata: From Open to Closed Consumption. *International Journal on Advances in Intelligent Systems*, 7(3&4), 662-688.
- Giova, G. (2011). Improving chain of custody in forensic investigation of electronic digital systems. *International journal of computer science and network security*, 11(1), 1-9.
- Glimm, B., Hogan, A., Krötzsch, M. et Polleres, A. (2012). OWL: Yet to arrive on the Web of Data? *Linked Data on the Web Workshop*.
- Hartig, O. (2009). Provenance Information in the Web of Data. *Linked Data on the web workshop*.
- Hartig, O., Bizer, C. et Freytag, J. C. (2009). Executing SPARQL queries over the web of linked data. *International semantic web conference*, 293-309.
- Hartig, O. et Zhao, J. (2010). Publishing and consuming provenance metadata on the web of linked data. *Provenance and annotation of data and processes*, 78-90.
- Hartig, O. et Zhao, J. (2012). Provenance Vocabulary Core Ontology Specification. Retrieved June 2011 from <http://trdf.sourceforge.net/provenance/ns.html>.
- Hartmann, J. et al. (2005). Ontology metadata vocabulary and applications. *Workshop on Ontology Patterns for the Semantic Web*, 906-915.
- Heath, T. (2008). How will we interact with the web of data? *Internet Computing, IEEE*, 12(5), 88-91.
- Heath, T. et Bizer, C. (2011). Linked data: evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1), 1-136.

- Heath, T., Hausenblas, M., Bizer, C., Cyganiak, R. et Hartig, O. (2008). How to publish linked data on the web tutorial. International Semantic Web Conference. Retrieved October 2012 from <http://events.linkedata.org/iswc2008tutorial/>.
- Hitzler, P. et Harmelen, F. v. (2010). A reasonable semantic web. *Semantic web*, 1(1-2), 39-44.
- Hogan, A., Harth, A., Umrich, J., Kinsella, S., Polleres, A. et Decker, S. (2013). Searching and browsing linked data with SWSE: the semantic web search engine. *Journal of Web semantics*, 9(4), 1-55.
- Insa, F. (2007). The admissibility of electronic evidence in court: fighting against High-Tech Crime - Results of a European Study. *Journal of Digital Forensic Practice*, 1(4), 285-289.
- Isaac, A. et Summers, E. (2008). Simple knowledge organization system primer. Retrieved October 2012 from <http://www.w3.org/TR/skos-primer/>.
- Isele, R., Umbrich, J., Bizer, C. et Harth A. (2010). LDspider: An open-source crawling framework for the web of linked data. International Semantic Web Conference, 29-32.
- Jentzsch, A. (2013). Describing and Comparing Datasets on the Web of Data.. Technical reports of the Hasso Plattner Institute software systems engineering at the University of Potsdam, 69-78.
- Jueneman, R. R. et LaPedis R. (2011). Solving the digital chain of custody problem. Trusted mobility solutions. White paper.
- Karnin, E. D., Greene, J. W. et Hellman, M. E. (1983). On secret sharing systems. *Information theory, IEEE transactions*, 29(1), 35-41.
- Keßler, C., D'Aquin, M. et Dietze, S. (2013). Linked data for science and education. *Semantic web*, 4(1), 1-2.
- Kessler, G. C. (2010). Judges' awareness, understanding, and application of digital evidence. Presented in partial fulfillment of the requirements for the degree of philosophy in Nova Southeastern University.

- Köhn, M., Eloff, J. H. P. et Olivier, MS. (2008). UML modeling of digital forensic process models (DFPMs). *Information security for South Africa*, 1-13.
- Krotoski, M. L. et Passwaters, J. (2011). Obtaining and Admitting Electronic Evidence. *United States Attorney's Bulletin*. 1-77.
- Kruse II, W. G. et Heiser, J. G. (2001). *Computer forensics: incident response essentials*. Indiana: Addison Wesley.
- Kuhn, D. R., Hu, Vincent C., Polk, W. T. et Chang, S.J. (2001). Introduction to public key technology and the federal PKI infrastructure, DTIC Document.
- Lagoze, C. et H. Van de Sompel (2003). The making of the open archives initiative protocol for metadata harvesting. *Library hi tech*, 21(2), 118-128.
- Losavio, M., Adama, M. et Rogers, M. (2006). Gap analysis: Judicial experience and perception of electronic evidence. *Journal of digital forensic practice*, 1(1), 13-17.
- McCalla, G. et Cercone, N. (1983). Guest Editors' introduction: approaches to knowledge representation. *Computer*, 16(10), 12-18.
- McGuinness, D. L. et Van Harmelen F. (2004). OWL web ontology language overview. *W3C recommendation*, 10(10).
- Mendelsohn, N. (2008). The self-describing web. Draft TAG finding, *W3C recommendation*.
- Menezes, A. J., et al. (1996). *Handbook of applied cryptography*. Florida: CRC press.
- Miller, F. P., Vandome, A. F. et McBrewster, J. (2009). *Advanced Encryption Standard*. USA: Alpha Press.
- Moreau, L. et al. (2011). The open provenance model core specification (v1.1). *Future generation computer systems*, 27(6), 743-756.
- Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International journal of artificial intelligence in education*, 10, 98-129.

- Omitola, T. et al. (2011). Tracing the provenance of linked data using void. International conference on Web intelligence, mining and semantics, (17).
- Pastor, S., Antonio, J., Mendez M., Javier F., Munoz R. et Vicente J. (2009). Advantages of thesaurus representation using the simple knowledge organization system Compared with Proposed Alternatives. Information research: An international electronic journal, 14(4), 1-16.
- Pearson, D. (2002). Presentation on grid data requirements scoping metadata and provenance. Workshop on data derivation and provenance.
- Perlman, R. (1999). An overview of PKI trust models. Network, IEEE, 13(6), 38-43.
- Poli, R., Healy, M. et Kameas, A. (2010). Theory and Applications of Ontology: Computer Applications. Chapter 1: the interplay between ontology as categorical analysis and ontology as technology.
- Prud'Hommeaux, E. et Seaborne A. (2008). SPARQL query language for RDF. W3C recommendation, 15.
- Quan, D. et Karger R. (2004). How to make a semantic web browser. International conference on World Wide Web, 255-265.
- Rajabi, E., Kahani, M. et Sicilia, M.A. (2012). Trustworthiness of linked data using pki. Proceedings of the World Wide Web conference.
- Request For Comment (RFC) (1999). Internet X.509 Public Key Infrastructure Certificate Management Protocols. Retrieved May 2012 from <https://tools.ietf.org/html/rfc4210>.
- Rijmen, V. et Daemen, J. (2001). Advanced encryption standard. National Institute of Standards and Technology: 19-22.
- Ringland, G. A. et Duce, D. A. (1988). Approaches to knowledge representation: an introduction. United Kingdom: Research Studies Press Ltd.
- Rivest, R. L., Shamir, A. et Adleman L. M. (1983). Cryptographic communications system and method, Google Patents.

- Rogers, M., Scarborough, K., Frakes, K. et Martin, C. S. (2007). Survey of law enforcement perceptions regarding digital evidence. *Advances in Digital Forensics III*, 242, 41-52.
- Salgueiro, F. A., Costa, G., Cataldi, Z., Lage, F. J. et García Martínez, R. (2005). Redefinition of basic modules of an intelligent tutoring system: the tutor module. VII workshop of researchers in computer science, 444-448.
- Sauermann, L., Cyganiak, R. et Völkel, M. (2011). Cool URIs for the semantic web. German Research Center for artificial intelligence, 1-15.
- Schatz, B. (2007). Digital evidence: representation and assurance. Presented in partial fulfillment of the requirements for the degree of philosophy in Queensland University of Technology.
- Schatz, B., Clark, A. et Mohay, G. (2004a). Rich event representation for computer forensics. *Asia Pacific industrial engineering and management system*, 1-16.
- Schatz, B., Clark, A. et Mohay, G (2004b). Generalizing event forensics across multiple domains. *Australian computer, network and information forensics conference*, 136-144.
- SCORM (2004). Advanced Distributed Learning. SCORM Overview. Retrieved June 2012 from <http://www.adlnet.org/>.
- Seifert, J. W. (2004). Data mining: an overview. *National security issues*, 201-217.
- Sherman, L. W., Gottfredson, D., MacKenzie, D., Eck, J., Reuter, P. et Bushway, S. (1997). Preventing crime: What works, what doesn't, what's promising: A report to the United States Congress, US Department of Justice, and Office of Justice Programs Washington.
- Smith, F. C. et Bace R. G. (2002). A guide to forensic testimony: The art and practice of presenting testimony as an expert technical witness. United Kingdom: Pearson Education.
- Turner, P. (2005). Unification of digital evidence from disparate sources (digital evidence bags). *Digital Investigation*, 2(3), 223-228.

- Van de Sompel, H., Lagoze C. et McGuinness, D. L. (2001). The open archives initiative protocol for metadata harvesting. Retrieved May 2011 from <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
- Van Harmelen, F. et McGuinness, D. L. (2004). OWL web ontology language overview. World Wide Web Consortium (W3C) Recommendation. Retrieved April 2011 from <http://www.w3.org/TR/owl-features/>.
- W3C. (2004). OWL Web Ontology Language Guide. Retrieved June 2011, from <http://www.w3.org/TR/owl-guide/>.
- W3C. (2006). XML Schema Datatypes in RDF and OWL. Retrieved April 2011 from <http://www.w3.org/TR/swbp-xsch-datatypes/>.
- W3C. (2014). RDF Schema 1.1. Retrieved May 2014 from <http://www.w3.org/TR/rdf-schema/>.
- W3C. (2015). RDFa Core 1.1 - Third Edition. Retrieved March 2015 from <http://www.w3.org/TR/rdfa-syntax/>.
- Wenger, E. (2014). Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. California: Morgan Kaufmann Publishers Inc.
- Yoo, C. S. (2005). Beyond network neutrality. Harvard journal of law and technology, 19(1).
- Yusoff, Y., Ismail R. et Hassan Z. (2011). Common phases of computer forensics investigation models. International journal of computer science and information technology, 3(3), 17-31.
- Zhao, J. (2010). Open Provenance Model Vocabulary Specification. Retrieved November 2012 from <http://open-biomed.sourceforge.net/opmv/ns.html>.
- Zhao, J., Bizer, C., Gil, A., Missier, P., et Sahoo, S. (2010). Provenance requirements for the next version of rdf. W3C Workshop RDF Next Steps.